TK 2000

MICROHOBBY

Ano I - N.º 12 - 1984 Cr$ 1 600,00

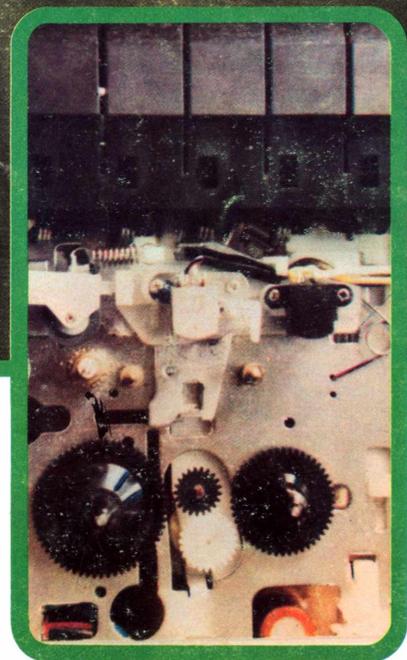
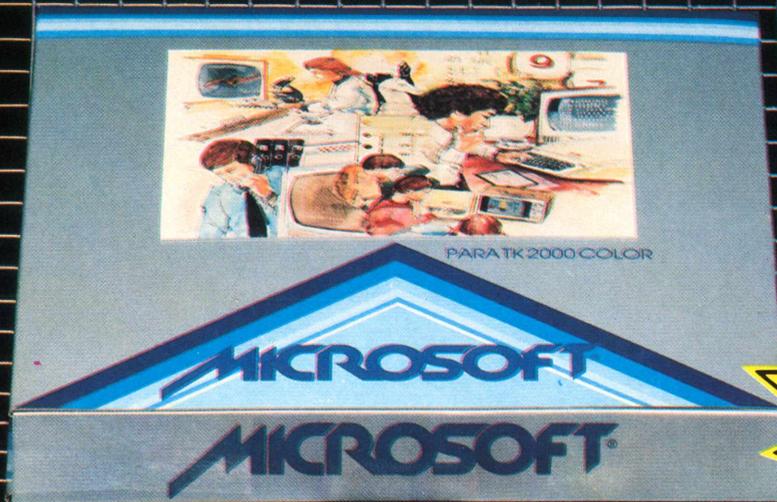
CARLA
(Freud Explica)
12 kBytes

I ♥ jung

Os Velhos Problemas Com o Gravador

Criptografia com o TK

Casando Impedâncias com o Programa Padre

Para Manaus, Boa Vista, Porto Velho, Rio Branco, Santarém, Altamira, Macapá, Est. de Rondônia (Via Aérea), Cr$ 2.080,00

Os melhores programas para você.
PARA TK 2000 COLOR
MICROSOFT
MICROSOFT
Garantia integral
MICROSOFT
MICROSOFT
MICROSOFT
MICROSOFT
A Microsoft tem 120 programas em fitas e disquetes à sua disposição.
São sistemas aplicativos para acompanhar e agilizar os negócios
de sua empresa. E também jogos eletrônicos para você e sua família
se divertirem muito. Todos especiais para TK-83, TK-85,
TK-2000, Apple II e compatíveis. E todos com a mesma qualidade
dos 100.000 programas já vendidos em todo o Brasil.
Procure o revendedor Microsoft mais próximo (se não encontrar os
programas Microsoft escreva para a Caixa Postal 54221 - CEP 01000-
S. Paulo-SP). Você encontrará os melhores programas da sua vida.
MICROSOFT
Sempre o melhor programa.

Campanhas nacionais de apoio à reserva

A reunião anual da Sociedade Brasileira para o Progresso da Ciência, realizada no mês de julho em São Paulo, surtiu inúmeros efeitos na área de informática. A reunião teve vários painéis sobre a mesma e a maioria versaram sobre a tão discutida reserva de mercado. Porém, a partir destes debates surgiram iniciativas mais concretas por um posicionamento mais claro com relação à questão.

Dessa forma, várias entidades — encabeçada pela SBPC — inauguraram o Movimento Brasil-Informática, contando com apoio da Abicomp, APPD — Nacional e Federação Nacional dos Enengenheiros. Após reunião entre os representantes destas entidades, seus participantes decidiram criar comitês regionais com o objetivo de alcançar todo o país numa mobilização completa da sociedade brasileira, para pressionar o Congresso Nacional na aprovação de uma lei que regulamente a política nacional de informática.

Em São Paulo, este movimento iniciou-se em maio deste ano na Universidade de São Paulo e a partir daí, ele seguiu para a Secretaria da Indústria, Comércio e Tecnologia quando o secretário Einar Kok encomendou à comissão de automação ligada ao Conselho Estadual de Política Industrial, Comercial e Agroindustrial, um documento que definisse a posição do Estado com relação ao assunto. Do documento resultaram algumas definições como o reconhecimento de que não há uma preocupação no Brasil nem estratégia de ocupação do mercado para uma política industrial. Com relação às telecomunicações, o documento afirma que o país foi retalhado entre as empresas multinacionais, resultando deste fato — segundo o documento — a eliminação da competição (um meio de servir o consumidor). É demonstrada, ainda, uma preocupação em salientar que a busca de autodeterminação tecnológica, através de uma posição madura e de competência, é essencial, não se esquecendo de procurar ver as necessidades para importar.

Após a iniciativa do secretário, as entidades se agruparam e passaram a atuar a nível de governo federal. Em Brasília, o movimento começa a esquentar, pois no Congresso Nacional existem cerca de cinco projetos tramitando e o Executivo, neste mês de agosto, enviará seu projeto de Lei propondo uma legislação para a política nacional de informática. O projeto do Executivo prevê a consolidação das diretrizes traçadas pela SEI e a reserva de mercado para a indústria brasileira de computadores.

Um outro movimento, que se instala a nível federal, é pela aprovação do anteprojeto que prevê o registro compulsório de software dentro de prazos que variam de seis (para programas nacionais) e um ano (para programas que venham do exterior). O anteprojeto foi encomendado pela SEI à OAB e prevê ainda o domínio público no caso dos softwares não registrados dentro destes prazos.

Simpósio Itautec

De 8 a 10 de julho realizou-se, no anfiteatro do Centro de Convenções Rebouças, simpósio promovido pela Itaútec, com o objetivo de mostrar a um público formado por empresários, professores, profissionais, jornalistas e estudantes, as perspectivas do uso do microcomputador em vários setores, perspectivas de desenvolvimento e a situação técnica, política e social atual.

Pouco antes da abertura oficial do evento, Olavo Setúbal, presidente do grupo Itaú, deu entrevista coletiva à imprensa. Na oportunidade ele voltou a defender a reserva parcial de mercado e o estabelecimento de uma política nacional de informática. Por reserva parcial de mercado, Setúbal entende que não podemos abrir totalmente o mercado às empresas multinacionais, uma vez que as empresas brasileiras não têm condições de competir em pé de igualdade com empresas estrangeiras de porte. Por outro lado, não podemos fechar completamente o mercado, pois

Olavo Setúbal presidente do grupo Itaú e Crodowaldo Pavan presidente da SBPC no evento Itautec.

corremos o risco de utilizarmos o mesmo tipo de modelo adotado nos países do Leste Europeu. Nesta situação, o Brasil teria que romper com uma série de compromissos de caráter internacional (como, por exemplo, patentes e direitos autorais) e montar um serviço de espionagem tecnológica. Ou, como ele mesmo exemplificou — mostrando os dois extremos — "ou compramos tudo ou nos apropriamos de tudo".

Teclado profissional para os TKs

É um complemento, desenvolvido pela Speed Eletro-Eletrônica, que fornece aos micros da linha TK aparência e as facilidades de um micro profissional. O teclado é composto por "switches" de acionamento mecânico, gabinete de fiberglass e foi dimensionado para comportar o micro, expansão e fonte de alimentação.

O teclado é do tipo profissional, padrão máquina de escrever, que permite a dinamização da entrada de dados no micro, tornando a digitação mais rápida. Os "switches" comercializados pela empresa podem ser usados também em teclados de equipamentos industriais, como controladores de processos industriais, comandos numéricos e digitais e ainda, segundo a empresa, em teclados de computadores profissionais.

Show-room da Victor presta serviços ao usuário

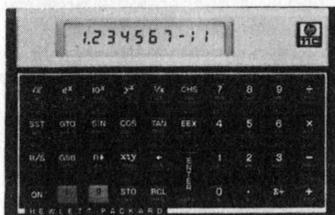
Israel Teixeira

Antena de recepção de sinais de satélite.

Expondo sua linha de produtos — Elppa Plus e o Elppa Junior — a Victor inaugurou um show-room que pretende prestar serviços aos usuários de seus micros, oferecendo software, cursos e literatura, além de também estar comercializando micros da Itaútec e todo seu suporte além dos monitores Elppa.

Uma das atrações do show-room Victor é uma parábola instalada na entrada, com o objetivo de demonstrar a recepção de sinais de satélite usando um micro Elppa na geração de caracteres. A empresa inicia suas atividades de apoio ao usuário logo no início de agosto, através de cursos e informações podem ser obtidas no próprio local, em São Paulo, pelo telefone (011) 872-4788.

Teclado da Speed para os micros da linha TK.

Duas calculadoras profissionais

Calculadoras-Hewlett Packard.

HP-12C e HP—11C são dois modelos de calculadoras que a Hewlett Packard apresentou na 11ª FUSE em junho último. As duas calculadoras atendem interesses de cálculos de duas categorias profissionais distintas: engenheiros e profissionais da área financeira.

A HP-11C possue um conjunto de funções científicas pré-programadas que permite cálculo de funções hiperbólicas, conversões entre graus e radianos, coordenadas retangulares e polares. Outra característica da 11-C são suas teclas inclinadas que reduzem a fadiga proveniente de uma utilização prolongada. A empresa fornece, juntamente com a calculadora, o manual do proprietário com inúmeros programas prontos como álgebra matricial, sistemas de equações lineares, resolução de triângulos e cálculo de distribuição qui-quadrado.

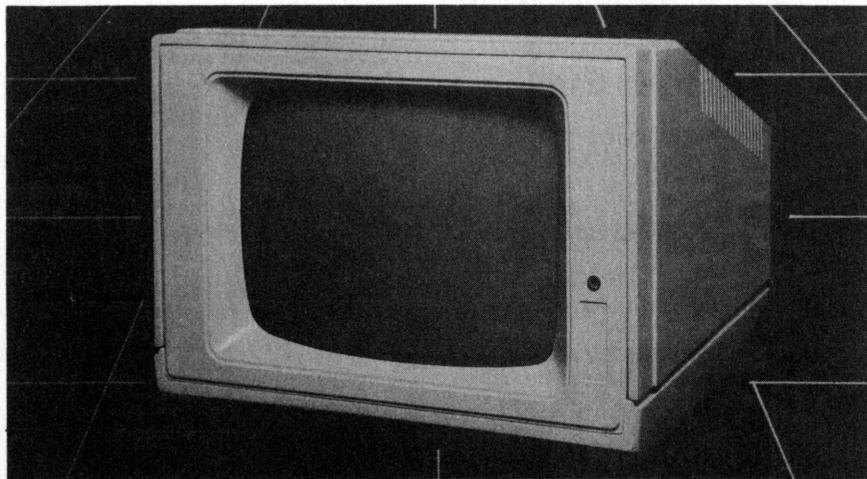
A HP-12C financeira programável permite a resolução de problemas de juros e de amortização, descontos de fluxos de caixa, cálculo de rendimentos de títulos e letras e depreciação (linear, exponencial, juros de dias singulares). A calculadora tem também revisão unitária ou contínua de programas (linha à linha, para frente ou para trás), alocação automática de memória, dois modos de separação das partes inteira e decimal (brasileiro e norte-americano) e sistema lógico (pilha automática com quatro memórias, recuperação de erros e manipulação da pilha operacional). Como a HP-11C, acompanha-a também, o manual do proprietário e o guia de aplicações.

Um monitor de vídeo para o TK-2000

Um monitor de vídeo profissional desenvolvido especialmente para o TK-2000 estará sendo oferecido aos seus usuários a partir de agosto próximo. O equipamento, fabricado pela Icomet-Indústria e Comércio, que atua no mercado de informática há mais de um ano com terminais de vídeo inteligentes para minicomputadores, foi desenvolvido inicialmente para micros da linha Apple. Porém, com o surgimento do TK-2000 no mercado, a empresa decidiu fabricar monitores compatíveis com ele.

O monitor, denominado Matriz-MX200, possue alta resolução gráfica, controles de brilho vertical e horizontal e tela anti ofuscante, uma característica que, segundo seus fabricantes, é única no mercado.

Segundo Carlos Eduardo Battendieri, diretor comercial da Icomet, a empresa decidiu fabricar o monitor para o TK-2000 devido a grande receptividade deste micro no mercado, graças — conforme afirmou — ao seu suporte técnico. "Além disso — continuou — este micro oferece possibilidades de ampliação de aplicações profissionais em função da relação entre preço e desempenho".

O monitor de vídeo passou pelos testes de desempenho da Microdigital e aprovado, será vendido a um preço que variará entre 45 a 50 ORTNs.

Micros TK e interface para impressoras

Os microcomputadores da linha TK têm agora uma interface paralela para se ligarem a impressoras do padrão Centronics como Emília, Mônica, Grafix, Epson etc. Acoplando a interface ao conector de expansão, os usuários de micros TK podem efetuar registros impressos de dados, programas e relatórios.

A Microdigital, além de lançar esta interface, está inaugurando sua nova fábrica na Barra Funda em São Paulo, com 3 mil m². A construção desta nova fábrica, segundo a empresa, tem o objetivo de acelerar o ritmo de produção, com o intuito de exportar tecnologia e serviços.

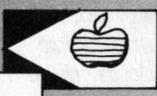
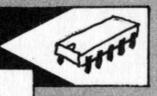
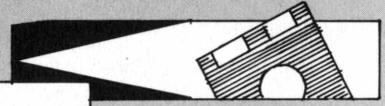
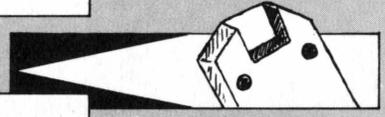
Palestras e exposição no 4º Semicro

De 27 a 31 de agosto a UFRJ fará realizar o seu 4º Semicro e a 3ª Microexposição de microcomputadores e periféricos nacionais. Promovido pelo Núcleo de Computação Eletrônica da UFRJ o evento conta com o apoio de várias entidades científicas como CNPq, Capes, SEI, Serpro, Abicomp e outras, além do patrocínio especial da Organização dos Estados Americanos — OEA.

Este evento, segundo seus organizadores, tem o objetivo de difundir o microcomputador como ferramenta básica na vida cotidiana e profissional. O Semicro se destina aos profissionais da área de computação, engenheiros e projetistas de toda as especialidades e à comunidade acadêmico científica da área de informática. Dentro de sua programação está a realização, em paralelo, do I Simpósio de Informática na Educação, do I Seminário de Microinformática na Universidade e o II Seminário de Microinformática na Administração Pública.

A 3ª Microexposição terá a participação de vários fabricantes de hardware, software e periféricos e será montada no hall do auditório do Centro de Tecnologia da UFRJ. Esclarecimentos maiores podem ser obtidos com a comissão organizadora no telefone (021) 290.3212.

Produtos para conservação de equipamentos eletrônicos

São fabricados por uma empresa do Rio de Janeiro — Aerojet — na forma de spray, com adequação à manutenção e limpeza de equipamento técnicos e industriais. Um dos produtos que merece destaque é o LC-150/Limpador de Contatos Eletrônicos com "freon" TF, indicado para limpeza dos sistemas que exigem continuidade elétrica como cabeças de gravação, discos de memória, circuitos integrados, contactores e transceptores. O LC-150 serve também para limpeza de fitas magnéticas cassete ou profissionais.

unitron
Ringo
MICRODIGITAL
PHILIPS
CMA
apple
ZIROK
EXATO
Matrix
MICROCRAFT
TEXAS
PRÓ
ELETRÔNICA
SOFTWARE
HARDWARE
SUPRIMENTOS
INSTRUMENTAÇÃO
COMPONENTES
ELETRÔNICOS
Venha conhecer nosso show room e participar ativamente das mais recentes e revolucionárias tecnologias a respeito dos microcomputadores. Além de fazer amigos, eventualmente poderão descobrir o que um Microcomputador poderá fazer por vocês ou pela sua Empresa.
Trata-se de uma perfeita e balanceada estrutura para representar à altura, as principais Empresas de Computadores tais como: Texas, Zirok, Microdigital, Dactari, Polymax, Unitron, Elebra, Aceco, Prológica, Apple II, Savage, Microcraft, CMA, Phillips, Ringo, Matrix...
Intellivision
CERTIFICADO DE REGISTRO NO CADASTRO DE FORNECEDORES DE SUPRIMENTOS DE INFORMÁTICA
SEI
Kaprom PROPAGANDA
RUA SANTA EFIGÊNIA, 568 - SP - FONE: 221-9055

EDITORIAL

Em meados do século XV, Portugal e Espanha entraram em uma disputa sobre a quem pertenceriam as terras descobertas no Novo Mundo. A disputa entre ambos foi resolvida por meio de um tratado, o famoso Tratado de Tordesilhas. Este tratado dividia o mundo em duas partes por meio de um meridiano e as terras a serem descobertas à sua esquerda seriam da Espanha e as que estivessem à sua direita, de Portugal. Tudo bem!

Tudo bem? Alguns países, especialmente a França, discordaram desta atitude e questionaram: "que direito têm dois países de dividir o Mundo como se fosse uma laranja?"

Que tem isso com a Informática? Muito. Atualmente divide-se entre os países desenvolvidos algo mais sutil e tão importante quanto as terras a serem descobertas: a tecnologia em Informática. Países como os Estados Unidos, Inglaterra, Japão, acham-se no direito de possuir este bem e acreditam que os países em desenvolvimento devem, no máximo, consumir pacotes prontos, onde pouco ou nada do conhecimento científico-tecnológico é realmente transferido. E tudo isso sem justificativa, ou com justificativas vagas ou até mentirosas, do tipo "nós temos a tecnologia porque soubemos desenvolvê-la. Vocês também não podem fazer isso porque são incapazes" ou "com a presença de nossas empresas em seus países vocês desfrutarão do benefício da nossa tecnologia".

Nós brasileiros, como outros membros de países em desenvolvimento, nos sentimos no dever de questionar: "que direito têm os países desenvolvidos de nos impor seus produtos e nos impedir de nos desenvolver?"

A melhor atitude que temos para provar que estamos certos é conquistar, com muita luta, um lugar neste Novo Mundo que não nos permitem habitar ou explorar, desenvolvendo uma tecnologia tão boa quanto a desenvolvida pelos "donos do Mundo" e sobretudo voltada para as nossas necessidades, visando a resolver nossos problemas e sem lançar mão de recursos fáceis que tapam o buraco agora, mas abrem-no novamente no futuro.

Alvaro

Índice

Capa: Cassiano Roda

Expediente

EDITOR
Álvaro A.L. Domingues
JORNALISTA RESPONSÁVEL
Ana Lúcia de Alcântara — Mt. 14495
NOTICIÁRIO
Ana Lúcia de Alcântara
ASSESSORIA TÉCNICA
Flavio Rossini
Wilson José Tucci
ANÁLISE E REDAÇÃO
Nancy Mitie Ariga
Carlos Eduardo Rocha Salvato
Gustavo Egídio de Almeida
Renato da Silva Oliveira
Roberto Bertini Renzetti
PRODUÇÃO GRÁFICA
José Carlos Sarkis

ARTE
Cassiano Roda
Eliana S. Queiroz Yoshihara
Fatima M. Rossini Gouveia
Osmère Sarkis
CORRESPONDENTES
New York — Flavio Rossini
PUBLICIDADE
Aurio Mosolino (supervisor)
Eduardo Garcia de Souza
CIRCULAÇÃO E ASSINATURAS
Marcia Regina Dominiqui
GERÊNCIA GERAL
Angel D. Zaccaro Conesa
DISTRIBUIÇÃO
Fernando Chinaglia Distribuidora S/A

MICROHOBBY é editada mensalmente por MICROMEGA PUBLICAÇÕES E MATERIAL DIDÁTICO LTDA., INPI 2992 Livro A

Endereço para correspondência:
Cx. Postal 54096 —
CEP 01296 — São Paulo, SP.
Para solicitar assinaturas (12 números) envie cheque nominal à MICROMEGA P.M.D. LTDA., no valor de Cr$ 19.800,00.
Tiragem desta Edição: 30.000 exemplares
NÚMERO 12

CARTAS DOS LEITORES

Apple, os Oitenta e outros

Sendo assinante da Microhobby desde o número 1, gostaria de acrescentar algumas coisas à argumentação da redação em relação aos leitores que têm reclamado das seções que falam de outros computadores:

a) alguns leitores, como é meu caso, foram introduzidos à computação pela compra de um TK e depois, embora mantendo o TK, adquiriram computadores Apple ou 80 compatíveis ou um TK 2000 para aplicações mais profissionais, permanecendo, portanto, interessado em ambas as linhas.

b) sendo a Microhobby uma revista basicamente didática, um pouco mais de cultura e informação na área não farão mal a ninguém, pois, embora sendo uma maquininha genial, os TK e seus compatíveis não constituem o universo da computação . . .

c) a apresentação de programas nos vários dialetos BASIC poderá, pelo conhecimento das instruções diferentes e suas equivalentes em cada tipo de máquina, facilitar a conversão de programas entre os vários computadores, o que dá uma margem de operação muito mais ampla. Afinal, encontramos programas em BASIC-TRS ou BASIC-Applesoft, que gostaríamos de converter para o TK, não é verdade?

d) vamos unir mais a família dos amantes da informática, sem criar radicalismo e sectarismos. Afinal, todos gostamos de computadores e algumas páginas sobre outras lógicas não devem ser motivo para polêmicas e desunião. Ao invés delas, porque não damos as mãos?

Orlando Holanda Lima
São Paulo — SP

Caro Orlando,

Agradecemos o seu posicionamento favorável à presença de outras máquinas nas páginas de nossa revista.

Sugestão

Gostaria que vocês editassem, juntamente com a listagem decimal, a listagem com Mneumônicos com as linhas de comentários sobre o que uma determinada instrução faz no programa. Também gostaria que fosse publicado um estudo detalhado das rotinas da ROM do TK e do modo de gravação do cassete (sua forma de onda, freqüência, nível de entrada e saída, etc.).

Cláudio Henrique de Castro
São José dos Campos — SP

Caro Cláudio,

Suas sugestões são válidas e já foram parcialmente implantadas na revista. Alguns programas em Linguagem de Máquina estão sendo publicados na forma que você sugere, principalmente

no nosso curso de Assembly Z80. As rotinas da ROM são muitas e seria necessário publicar-se um livro a respeito. Mas, por outro lado, existe a seção dissecando, onde eventualmente publicamos alguma coisa sobre o assunto.

Mais sugestões

Escrevo esta carta para transmitir-lhes minha admiração pelo conteúdo desta revista, o que eu, sinceramente, não esperava encontrar quando fiz meu pedido de assinatura ao ver o número zero.

Tendo feito minha assinatura apenas para colaborar com uma nova publicação na área de informática, logo constatei que a minha escolha tinha sido mais do que certa. Espero assim que vocês continuem nesta linha que eu imputo de grande relevância para o desenvolvimento dos principiantes e mesmo dos já iniciados, numa área que não tinha grandes opções, antes do aparecimento da Microhobby.

Espero que vocês apresentem mais programas e dicas sobre aplicativos, sem se desviarem dos micros compatíveis com o TK. Sobre as sessões sobre o Apple e TRS-80 pediria que ficassem como complemento e fossem apresentados começando pelo mais simples.

Francisco J. Pazino Beceiro
São Paulo — SP

Caro Francisco,

Se você tem observado nossa revista, deve ter notado que tem aparecido bons aplicativos (veja o Dinheiro no Tempo, por exemplo) e que as sessões para o Apple e o TRS-80 não têm prejudicado o espaço destinado ao TK 83 e seus compatíveis. Desta forma, acreditamos que ainda somos merecedores da sua confiança quanto a manutenção da linha. É verdade que estamos dedicando algumas páginas a outras máquinas, mas isto vem sendo feito de maneira muito discreta e paulatinamente de tal modo que nosso leitor tradicional não sofra com a mudança.

Conteúdo

Gosto demais desta revista que os senhores publicam. Tanto que estou respondendo à pesquisa que vem sendo publicada em suas páginas desde o número 6. E claro que não posso pensar só em minhas necessidades e nas de meus filhos. Acho que, para uma clientela variada, a revista precisa ser também variada como é. Há matérias que a gente não lê, mas que, me parece, outros aproveitam muito. O que mais aproveitamos aqui em casa (eu, meus três filhos e minha esposa) são: as aulas de BASIC e Assembly, os programas, as informações sobre informática, etc. Desejo um dia (tempo . . .), organizar um programa de estatística para pesquisas no campo educacional e social. (. . .)

Danilo Gandin
Porto Alegre — RS

Caro Danilo,

Agradecemos os elogios e comentários favoráveis à nossa linha editorial. Procuraremos, à medida do possível, tornar esta revista ainda melhor. Assim como você, uma boa parte dos nossos assinantes respondeu a pesquisa que publicamos em alguns de nossos números. Desta forma, teremos uma idéia clara do que pensam nossos leitores e, de posse destes dados, pensar em como melhorar a revista de modo a atender à maioria dos leitores.

DESGRILANDO

*Existem algumas dúvidas que têm aparecido freqüentemente em nossa redação e merecem um esclarecimento por parte da **Microhobby**. Fizemos uma compilação e as generalizamos, para atender a todos os leitores com dúvidas semelhantes.*

Meu programa não funciona. Digitei tudo certo, mas mesmo assim ele continua dando erro. Há erro na revista?

Na maioria dos casos, não há erro na revista, uma vez que sempre fazemos uma revisão rigorosa dos programas e o que aparece na revista é uma cópia exata de uma listagem de um programa que está rodando perfeitamente em um de nossos computadores.

O que pode estar ocorrendo é:

1) **Erro na digitação.** Isso é o que leva mais pessoas a nos consultar. Assim sendo, tome cuidado com a digitação de comandos, instruções e funções. No TK e seus compatíveis, não podemos digitá-los letra por letra. Os comandos e instruções podem ser obtidos quando o cursor está em **K** ou **L** + SHIFT (comandos com STOP, LPRINT, ETC.). As funções são obtidas digitando-se SHIFT + NEW LINE (ao mesmo tempo). Deste modo, obteremos o cursor **F**, que indica que o computador está no modo FUNCTION e podemos digitar a função que quisermos (SIN, TAN, NOT, ETC.).

Importante: PI também é uma função (representada no teclado por π). Uma boa norma é só digitar letra por letra o que não aparecer no teclado.

2) **Erro na interpretação ou operação do programa.** Isso ocorre quando o texto não é lido em sua integridade ou aparecem dúvidas em sua leitura. É aconselhável digitar-se o programa apenas após a leitura detalhada do texto e sem que haja dúvidas. O principal problema que ocorre aos nossos leitores é perder um programa ou dados, digitando-se RUN quando devia-se digitar GOTO um determinado número de linha ou RAND USR um determinado endereço.

3) **Listagens com ambiguidades:** Podem aparecer pontos onde você pode se confundir com alguma letra. Isso ocorre algumas vezes em programas em Linguagem de Máquina onde o **B** pode ser confundido com o **8** (oito). Se você acha que digitou o programa certo, conferiu a listagem e ainda assim tem problemas, é conveniente procurar por ambigüidades (confusão de letras, números e símbolos gráficos).

4) **Erro na revista:** Se você digitou o programa corretamente, procurou todas as possíveis ambigúidades, conferiu e reconferiu e mesmo assim o programa não funciona, comece a suspeitar da revista. Embora tenhamos todo o cuidado com a digitação, teste, preparação e conferência do programa, errar é humano. Embora isso seja raro, não é impossível. De qualquer forma, todas as vezes que apareceu um erro na revista, ele foi prontamente descoberto, a tempo de prepararmos uma errata. Se, por acaso, encontrar um erro que ainda não foi corrigido, escreva-nos, que procuraremos, por meio da seção *Cartas dos Leitores,* sanar o problema.

O que acontece se eu modificar uma das linhas do programa?

Experimente! Se você passou um dos programas da nossa revista ou de outra ou ainda de um livro, grave-o para evitar perdas, e faça um teste. Não tenha medo pois isso não prejudica o computador. O máximo que pode acontecer é você perder o programa ou o computador entrar em "crash", ou seja, não aceitar mais nenhum comando, mesmo BREAK. Isso não é catastrófico, pois ele sempre voltará ao normal quando o desligarmos e o ligarmos novamente.

Qualquer programa publicado ainda permite alterações e adaptações. Por isso não se acanhe em fazê-las. É importante, contudo, fazer estas modificações com um objetivo em mente (melhorar o algoritmo, melhorar a apresentação da tela, acrescentar mais algumas rotinas ou simplesmente "ver o que acontece"). Uma vez feita a modificação, anote-a para uso futuro, teste-a e anote o resultado, mesmo que seja negativo (assim você saberá o que **não** deve fazer!).

A fita com tal programa não entra. Que devo fazer?

A maioria dos problemas com a entrada de programas em fita no computador se dá por problemas de ajuste de volume e tonalidade do gravador, sujeira na cabeça, mal manuseio da fita, desajuste da cabeça do gravador e, por *último,* um defeito na fita. Por este motivo, temos publicado regularmente artigos visando sanar estes problemas. ○

DISSECANDO: FOR NEXT

Carlos Eduardo Rocha Salvato

INICIANTES

A forma pela qual se utiliza a instrução FOR/NEXT normalmente é: FOR variável contadora = valor inicial TO valor final (STEP incremento). NEXT variável contadora

O principal objetivo das instruções FOR/NEXT é executar a repetição de instruções em um programa.

Após digitar a instrução ou grupo de instruções que você quer que o programa repita devem ser colocadas entre o FOR e o NEXT.

Vamos analisar o seguinte programa:

```
10 LET B=1
20 FOR A=1 TO 10
30 PRINT B
40 LET B=B+1
50 NEXT A
60 STOP
```

nº da linha	comentários
10	Atribui o valor 1 para a variável B
20	Inicia a contagem de 1 até 10
30	Imprime o valor de B
40	Soma 1 no valor de B
50	Soma 1 ao valor de a e verifica se acabou a contagem
60	Para o programa

O que nos interessa realmente, são as linhas 20 e 50. Na linha 20 é atribuído à *variável-contadora A* o valor inicial no caso, 1. As instruções das linhas 30 e 40 são executadas e quando o programa chega na linha 50 NEXT, que faz duas coisas:

1. é somado 1 à variável contadora;

2. é verificado se o valor da *variável-contadora* é maior que o valor final no caso, 10. Se for maior, ele passa para próxima linha do programa, caso contrário, o programa volta à linha onde existe a instrução FOR.

Tente digitar o seguinte programa:

```
10 FOR A=1 TO 10
20 NEXT A
30 PRINT A
```

Após a análise do primeiro programa esta tarefa se torna muito fácil: quando a variável A tinha o valor 10, o micro, quando executou a linha 20, somou 1 à variável. Como ela era maior do que o valor final, estabelecido pela instrução FOR, o programa passou para a próxima linha, que ordenava que A fosse impressa.

STEP — um complemento para a instrução FOR — NEXT

O STEP é usado como incremento.

Veja o exemplo a seguir:

```
10 FOR A=2 TO 20 STEP 2
20 PRINT A
30 NEXT A
```

Digite RUN e veja o resultado. O TK imprimiu valores de 2 até 20, seqüencialmente, de dois em dois. Experimente agora alterar a linha 10 para:

```
10 FOR A=2 TO 20 STEP 3
```

O micro imprimirá os números entre 2 até 20, mas imprimindo agora, um número sim, dois não.

Usando o FOR — NEXT como tempo de espera

Muitas vezes queremos imprimir uma certa mensagem no vídeo e deixá-la durante algum tempo. Existe uma função especialmente feita para isto: a função PAUSE.

Rode o programa:

```
10 PRINT AT 10,6;"MICROHOBBY"
20 PAUSE 240
30 CLS
```

Você percebeu o efeito que o PAUSE causou: o vídeo piscou duas vezes. Isto pode ser evitado usando o FOR — NEXT como tempo de espera.

Se tirarmos a linha 20 e digitarmos as duas linhas que seguem, você perceberá como o efeito do FOR — NEXT é bem melhor.

```
20 FOR A=1 TO 240
25 NEXT A
```

Se o vídeo ficou cinza durante alguns segundos, provavelmente isto se deve ao fato de seu computador estar no modo FAST.

Se o seu TK estiver no modo FAST passe-o para o modo SLOW e rode o programa novamente.

O efeito foi bem melhor, afinal de contas o vídeo não piscou nenhuma vez.

Lembre-se: sempre que, quando você for usar o FOR — NEXT como tempo de espera, deixe seu microcomputador no modo SLOW.

MINÚCIAS

Simulando o FOR — NEXT

Existe uma restrição no TK, que limita o número de *variáveis-contadoras* em 26. Isto se deve ao fato do TK só aceitar uma letra como *variável-contadora*.

Se algum dia você for traduzir um programa de um outro computador qualquer, que entenda a linguagem BASIC, e encontrar alguma parte do programa que tenha o seguinte:

```
10 FOR AC=6 TO 89 STEP 5
20 PRINT AC
30 NEXT
```

Para traduzir isto, você terá duas opções:

A primeira delas seria a seguinte:

```
10 FOR A=6 TO 89 STEP 5
20 PRINT A
30 NEXT A
```

Mas tome cuidado para ver se a variável que você substituiu já não existe no programa.

A segunda já é um pouco mais complicada, mas também funciona. Digite a listagem a seguir:

```
10 LET AC=6
20 PRINT AC
30 LET AC=AC+5
40 IF AC>89 THEN GOTO 60
50 GOTO 20
60 STOP
```

Na tabela a seguir, nós explicaremos o programa anterior.

A instrução FOR/NEXT usada para fazer gráficos

A instrução FOR/NEXT é muito prática para ser usada na construção de gráficos, porque a variável contadora pode funcionar como os pontos do eixo X e, além disso, também pode ser usada na própria função cujo gráfico será construído. Veja o exemplo seguinte:

```
10 FOR A=0 TO 63
20 LET X=A
30 LET Y=22+20*SIN (X/32*PI)
40 PLOT X,Y
50 NEXT A
60 STOP
```

Se fosse feita uma tabela dos pontos do gráfico, ela seria a seguinte:

X	Y
0	22
1	23.960343
2	25.901805
3	27.805694
4	29.653669
5	31.427935
6	33.111405
7	34.687866
8	36.142136
9	37.460209
ETC	ETC
:	:

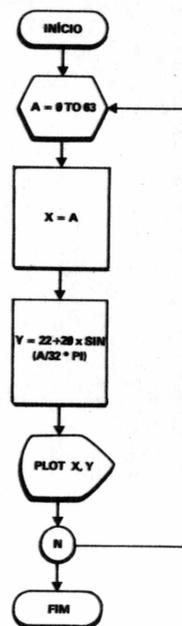
Tome cuidado para não ultrapassar os limites do vídeo senão será acusado o erro B.

Como representar o FOR/NEXT no fluxograma

A instrução FOR no fluxograma, sempre é ligada a instrução NEXT. A representação dessas duas funções no fluxograma é feita a seguir:

Observação: A letra N no centro do círculo é a abreviação da palavra NEXT e pode ser omitida.

Tente fazer o fluxograma do último programa apresentado.

Logo após você acabar, confira com o fluxograma a seguir:

LINGUAGEM DE MÁQUINA

Como já foi explicado em números anteriores, para inserirmos um programa em linguagem de máquina no TK, nós precisamos da ajuda de um pequeno programa chamado MONITOR, cuja listagem está a seguir:

```
10 REM N ZEROS
20 PRINT "ENDERECO INICIAL ?"
30 INPUT A
40 PRINT A
50 PRINT "ENDERECO FINAL ?"
60 INPUT B
70 PRINT B
80 FOR C=A TO B
90 SCROLL
100 PRINT C,
110 INPUT A$
120 PRINT A$
130 LET D=16*CODE A$+CODE A$(2)
-476
140 POKE C,D
150 NEXT C
160 STOP
```

Na linha 1 REM devem existir N pontos, onde N é um número que será determinado sempre antes do programa em linguagem de máquina. Os endereços inicial e final também serão →

fornecidos antes do programa. (Lembre-se: só digite os números que estão embaixo da palavra CÓDIGO.)

Aplicação prática do FOR/NEXT compilado

O termo compilar quer dizer o seguinte: traduzir um programa-fonte para uma linguagem de máquina de um determinado computador.

No nosso caso o FOR/NEXT seria o programa-fonte e o TK seria o computador.

Existem várias maneiras de se compilar as instruções FOR/NEXT. Nós veremos duas maneiras que podem ser usadas quase sempre.

A primeira delas é a seguinte:

```
   LD B, número de vezes que as instruções
                          serão executadas
L1 P
   R
   O
   G
   R
   A
   M
   A

   DJNZ, L1

   RET
```

O esquema anterior mostra como usar o registro B para compilar o FOR/NEXT.

Você pode usar o FOR/NEXT envolvendo todo o programa ou apenas uma parte dele.

A primeira instrução antes das instruções que vão ser repetidas deve ser aquela que carrega o número de vezes que as instruções posteriores serão repetidas, no registro B. Logo após esta instrução devem vir as que vão ser repetidas e, no final, deve vir a instrução DJNZ, L1, onde L1 é a posição onde começa o seu programa ou a sub-rotina que será repetida várias vezes.

A instrução DJNZ, L1 equivale ao NEXT e também faz duas coisas:

1. decrementa o contador B.
2. se B não for Ø ele volta para L1, caso contrário, continua o programa.

Lembre-se: L1 é o label onde começam as instruções que vão ser repetidas.

Caso o seu programa já utilize o registro B, você poderá utilizar outros registros. Para compilar um FOR/NEXT usando outro registro, nós teremos apenas que substituir a instrução DJNZ, L1 por duas outras instruções.

Fazendo um pequeno esquema ficaria assim:

```
     LD  registro, número de vezes que
         instruções serão executadas
L1   P
     R
     O
     G
     R
     A
     M
     A

     DEC registro
     JR NZ, L1
     RET
```

Nós precisamos fazer esta alteração, porque a instrução DJNZ só decrementa o registro B.

Observação: Se você for compilar um FOR/NEXT, cujo valor de vezes que as instruções serão repetidas for maior que 255, você deverá utilizar um par de registros no lugar de apenas um.

Compilando o FOR/NEXT

A seguir serão apresentados dois programas: um em BASIC e outro em ASSEMBLY (linguagem de máquina). Sendo que, o programa em Assembly teve a sua origem no programa em BASIC. Os dois programas irão inverter o vídeo cinco vezes. Repare a diferença da velocidade dos programas:

a) versão BASIC

```
  10 FOR A=1 TO 5
  20 FOR I=0 TO 21
  30 FOR J=0 TO 31
  40 PRINT AT I,J;
  50 LET L=PEEK (PEEK 16398+256*
PEEK 16399)
  60 IF L<128 THEN GOTO 130
  70 IF L>=128 THEN GOTO 150
  80 PRINT AT I,J;CHR$ L
  90 NEXT J
 100 NEXT I
 110 NEXT A
 120 STOP
 130 LET L=L+128
 140 GOTO 80
 150 LET L=L-128
 160 GOTO 80
```

b) versão ASSEMBLY

N = 22
Endereço Inicial = 16514
Endereço Final = 16535

Endereço	Mnemônicos	Códigos
16514	LD A,5	3E Ø5
16516	LD DE,(4ØØC)	ED 5B ØC 4Ø
1652Ø	INC DE	13
16521	LD BC, ØØ21	Ø1 21 ØØ
16524	LD H,D	62
16255	LD L,E	6B
16526	ADD HL,BC	Ø9
16527	LD BC, Ø2B5	Ø1 B5 Ø2
1653Ø	LDIR	ED BØ
16532	DEC A	3D
16533	JR NZ, −18	2Ø ED
16535	RET	C9

Observação: Para aumentar o número de inversões de vídeo, digite o seguinte:

POKE 16515, número de vezes desejada

O programa a seguir só tem a versão Assembly, pois ele é a versão compilada da instrução SCROLL.

N = 24
Endereço Incial: 16514
Endereço Final = 16538

Endereço	Mnemônico	Códigos
16514	LD D, Ø5	16 Ø5
16516	LD HL, (4ØØC)	2A ØC 4Ø
16519	LD B, 17	Ø6 17
16521	INC HL	23
16522	LD A,(HL)	7E
16524	CP 76	FE 76
16526	JR NZ, +6	2Ø Ø6
16528	DJNZ, −7	1Ø F8
16530	DEC D	15
16531	JR NZ, −15	2Ø fØ
16533	RET	C9
16534	ADD A, 8Ø	C6 8Ø
16536	LD (HL),A	77
16537	JR, −16	18 EF

Esta sub-rotina é muito útil, apesar dela fazer quase a mesma coisa que o SCROLL. As principais diferenças são:

1) esta sub-rotina executa cinco SCROLLs seguidos;

2) quando é feita uma limpeza de vídeo (CLS), ela é imediata e não demora como a instrução SCROLL do TK.

Observação: Para alterar o número de SCROLLs realizados, digite:

POKE 16515, número de SCROLLs desejados

Para testar a diferença da velocidade do SCROLL e da sub-rotina que simula o SCROLL, digite o seguinte programa:

```
   1 REM Y▌ GOSUB ?£RND<▀5 ??▀▀▙
▪ GOSUB ▩X4 GOSUB TAN
  10 PRINT AT 20,0;"     MICROMEG
A       MICROHOBBY"
  20 RAND USR 16514
  30 GOTO 10
```

A linha 1 REM contém a sub-rotina que simula o SCROLL.

Agora apague a linha 1 e digite as seguintes linhas:

```
  10 PRINT AT 20,0;"     MICROMEG
A       MICROHOBBY"
  20 FOR A=1 TO 5
  25 SCROLL
  27 NEXT A
  30 GOTO 10
```

OS VELHOS PROBLEMAS COM O GRAVADOR

Tanios Hamzo

OS DEZ MANDAMENTOS:

I — Não exporás tua fita gravada a nenhum campo magnético.
II — Não permitirás que se magnetize o cabeçote de teu gravador.
III — Não tocarás em tua fita nem no cabeçote de teu gravador em vão.
IV — Não apagarás em teu computador aquele programa sem que estejas certo de ter gravado-o para toda a eternidade.
V — Não permitirás programa sem cópia e a fará em duas fitas diferentes, pois assim terá o seu original uma cópia; e a tua cópia um original.
VI — Honra a teu gravador e a tua fita; e também o programa nela gravado; e também teu computador; e cuida destes como a ti mesmo, porque são tua ferramenta e teu trabalho.
VII — Não gravarás em fita nenhuma que não seja virgem.
VIII — Manterás imaculado o cabeçote de teu gravador.
IX — Não usarás fita incompatível com teu gravador; nem gravador incompatível com tua fita.
X — Praticarás sempre o uso do BEM de consumo apropriado e combaterás o MAU produto, para que não te decepciones; nem a teu próximo.

É quase impossível que você nunca teve problemas em carregar programas gravados em fita cassete no seu computador. Há histórias do arco e da velha, contando como importantes arquivos e grandes programas foram permanentemente perdidos, para a felicidade de seus usuários. O problema, porém não está no equipamento e sim na forma correta de usá-lo e ajustá-lo.

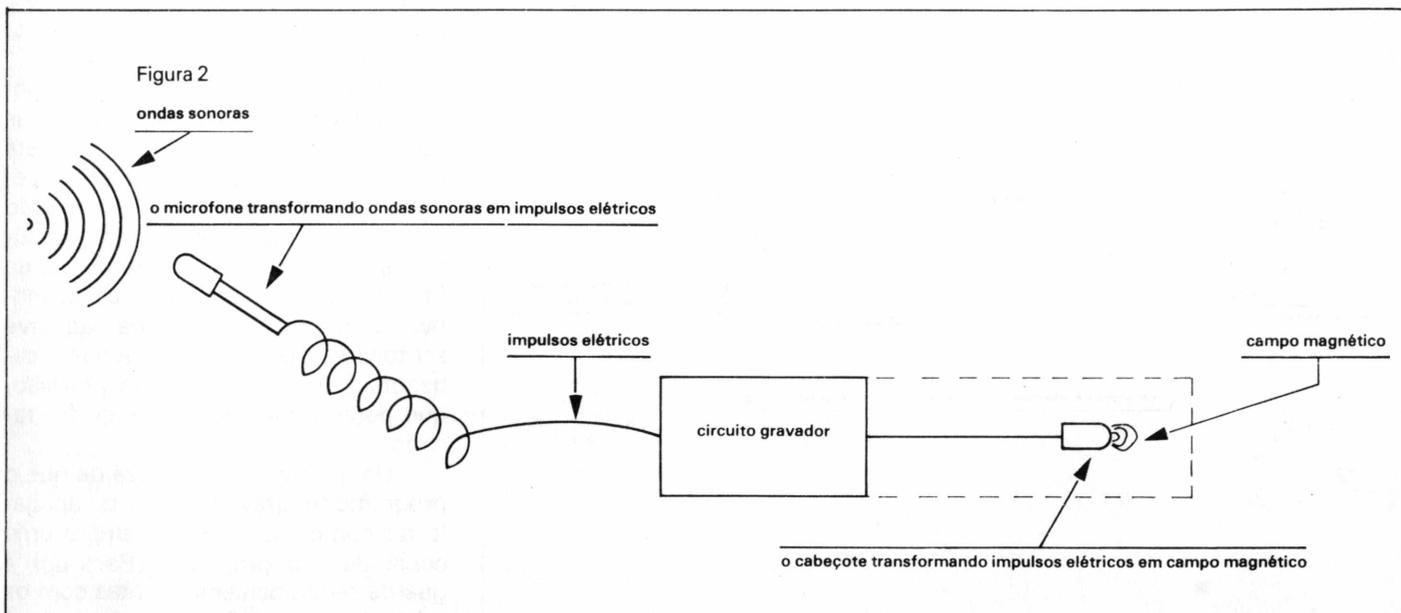
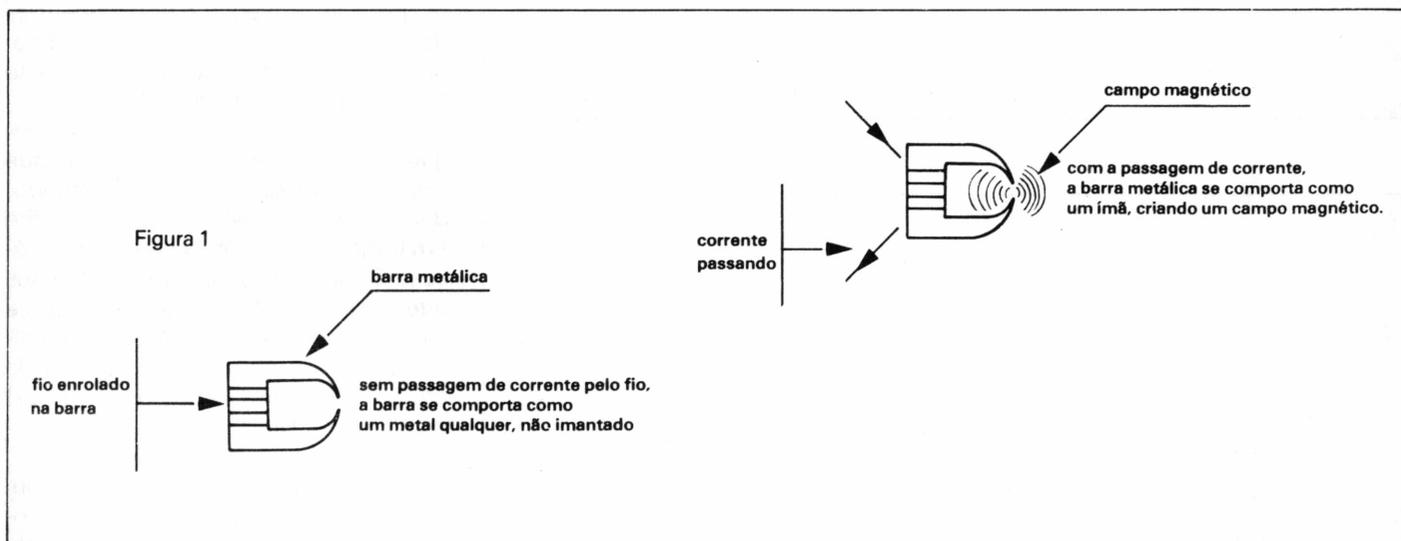
Descrevemos aqui alguns mandamentos que englobam qualquer caso, desde que estejam o gravador, o computador e a fita em bom estado de funcionamento. Primeiramente darei uma idéia de como funciona a gravação magnética.

O princípio de funcionamento

A figura 1 mostra o princípio de como funciona o cabeçote do gravador, criando um campo magnético concentrado numa fenda estreita. A intensidade deste campo é proporcional à corrente que circula no enrolamento (o fio enrolado, também chamado bobina), que por sua vez é proporcional ao sinal sonoro que está se gravando. Acompanhe pela figura 2 o caminho que o sinal sonoro de uma voz percorre até ser transformado em campo magnético por meio de transdutores (tradutores de sinais elétricos em outros e vice versa).

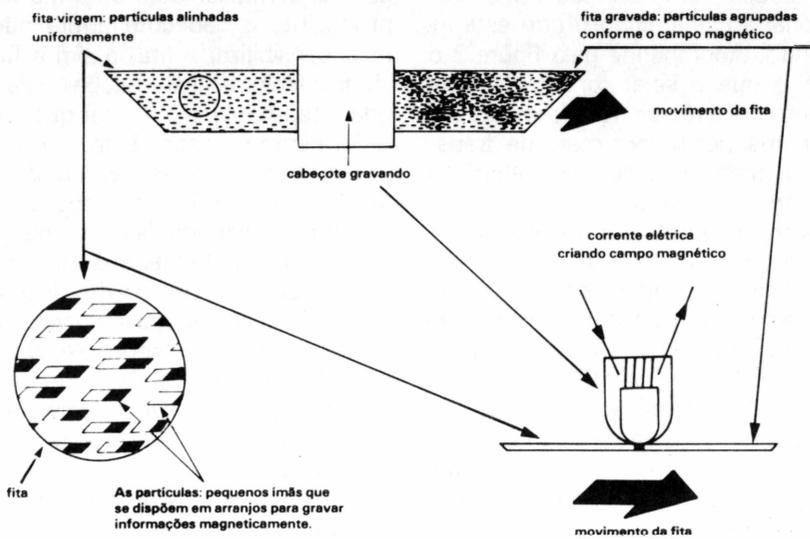
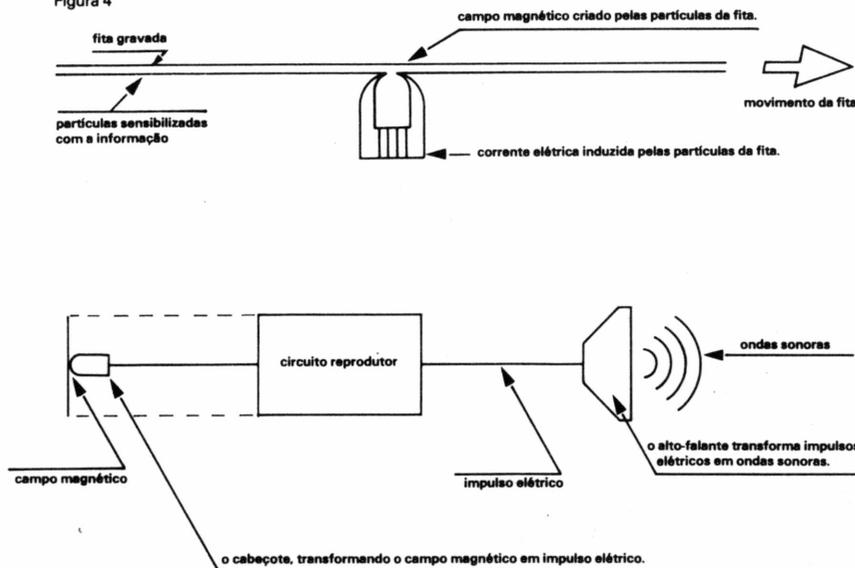
A fita magnética é composta por minúsculas partículas de um material magnético que podem ser arranjadas em diferentes posições na presença de um campo magnético, como o do cabeçote. A figura 3 ilustra esta operação. Pode-se comparar a "escrita" do cabeçote na fita como sendo um escriba fenício escrevendo num rolo de papel, que vai passando por ele à medida em que escreve.

A fita magnética é "escrita" magneticamente pelos cabeçotes com "escrita fina", ocupando regiões bem definidas e nítidas, ficando desta forma até que haja a presença de um outro campo magnético e altere as informações aí contidas. Quando a fita for reproduzida, o cabeçote agora não irá mais sensibilizar a fita, e sim a fita irá lhe transmitir as informações nela contida , reproduzindo o sinal que foi anteriormente gravado. Esta "leitura" se processa quando as partículas imantadas da fita recriam no cabeçote uma corrente similar àquela usada para sensibilizar a fita. Perceba a importância da precisão com que a fita deverá estar alinhada ao cabeçote e a velocidade com que ela desliza sobre ele. No caso de programas gravados, tudo se passa para o gravador e a fita como se fosse o sinal do computador uma onda sonora comum.

A figura 4 mostra o que acontece na reprodução da fita.

Cuidados específicos

Leia os dez mandamentos (veja o Quadro) para você não perder um programa. Estes "mandamentos" valem também para fitas contendo música ou qualquer outra informação. Estes cuidados apenas parecem exagerados. O conteúdo de uma fita é seriamente deteriorado se esta for exposta a um campo magnético, pois por natureza, é construída para ser sensível à sua presença. Embora estejamos sendo constantemente bombardeados por campos magnéticos, a aproximação de uma fita gravada a qualquer motor elétrico (ônibus elétrico, ventilador, etc.), transformadores (fonte de alimentação, TV, estabilizador de tensão, etc.), alto-falantes (do próprio gravador, caixas acústicas, o da TV, etc.), ou de algum outro imã permanente (chave de fenda, etc.) pode acarretar algum dano. A própria fita é magnética, composta de inúmeros pequenos imãs. Para quem já esqueceu das experiências do 1º grau, deslizando-se um imã sobre um corpo metálico imantável, este torna-se também um imã. É exatamente isto que faz com que o uso prolongado de um gravador faça com que seu cabeçote acabe ficando imantado, devido ao constante deslizar da fita (imantada) sobre ele (de metal imantável). Quando isto acontece, dizemos que o cabeçote ficou magnetizado e é inevitável que isto aconteça normalmente com o uso. O uso de fitas de cromo (Cr_2), de Ferro-Cromo (FeCr) ou fitas tipo Metal em gravadores comuns, que só aceitem fitas de Óxido de Ferro (as comuns, cor de ferrugem — Fe_2O_3) é impróprio e acelera o processo de magnetização do cabeçote, além de aumentar consideravelmente o desgaste deste por terem caráter abrasivo em cabeçotes convencionais.

Não toque o dedo ou qualquer objeto no cabeçote do gravador a menos que você saiba muito bem o que está fazendo. Fica sempre presente nos dedos um pouco de suor, que tem ação corrosiva no cabeçote, além de gordura e sujeira, que grudam partículas da fita, desgastando-a. Pelo mesmo motivo, a fita propriamente dita não deve ser tocada. Tenha em mente que o cabeçote é um dispositivo de precisão, que pode perder seu ajuste se for tocado.

Uma dica: tenha certeza de que o programa foi gravado antes de apagá-lo do computador. Faça sempre uma cópia de seu programa (Back-up) e guarde separadamente as fitas com os originais das fitas Back-up. Assim terá

a garantia de que, se acidentalmente um campo magnético deteriorar um grupo, o outro estará salvo. Costumo fazer isto usando fitas de marcas diferentes e constantemente verifico se estão em ordem, carregando o programa ora do original, ora da cópia.

Calor excessivo, pó e queda fazem mal a qualquer um, ainda mais se for em aparelhos de precisão relativa, como gravadores, computadores e fitas magnéticas. Não sou nem quero fazer de você um supercuidadoso, apenas dou atenção ao que acho que merece.

Os gravadores são munidos de um segundo cabeçote, usado para apagar e polarizar a fita antes de ser regravada. Porém no caso de programas, esta "limpeza" prévia muitas vezes é insuficiente. Recomendo usar fitas virgens para gravar programas, mas se isto não for possível apague a fita muito bem antes de regravar, usando algum apagador de fitas, do tipo "Twin-Go" ou, em último caso, desgrave no próprio gravador, desligando a entrada do microfone e gravando várias vezes um som "branco".

Mantenha sempre limpo o cabeçote do gravador, usando fitas de qualidade razoável, pelo menos. Fitas de má qualidade, as que não tem marca (na maioria, as que já vem gravadas), geralmente sujam muito o cabeçote. Também o fazem fitas velhas ou expostas em demasia ao calor (sol, por exemplo). Não use fitas amassadas ou enrugadas nem emende fitas partidas. UMA FITA BOA E NOVA É MAIS BARATA DO QUE A INSTALAÇÃO DE UM CABEÇOTE NOVO. Não se iluda: Fita adesiva e partículas podem acabar com o cabeçote. "Água mole em pedra dura . . ."

Os tipos

Use somente gravadores e fitas compatíveis entre si. Os tipos de fita magnética que existem são (em ordem cronológica e indo do tipo aceitável ao excelente, independendo do fabricante):

TIPO I — Óxido de Ferro (Fe_2O_3)

São as fitas normais, muito comuns e baratas. As fitas com baixo nível de ruído são melhores para uso em computadores. Nem sempre as mais caras são as melhores, e se puder, teste várias marcas e opte pela que der um som mais estridente e agudo. O que pode não ser muito bom para música pode ser excelente para se gravar programas. No mercado brasileiro existem boas fitas, inclusive algumas destinadas especialmente ao usuário de computador.

TIPO II — Cromo (Cr_2)

As fitas de Cromo reproduzem sons agudos muito melhor do que as de Óxido de Ferro, além de dificilmente deteriorarem o conteúdo, por serem menos sensíveis a campos magnéticos externos. Outra vantagem é a sua eficiência, gravando os sinais a um nível mais alto. Eu a considero a mais apropriada para a gravação de programas, embora seja mais cara do que as do tipo I e exija gravadores compatíveis com fitas de Cromo. Há várias marcas nacionais e importadas e a pior delas é melhor do que a melhor das do tipo I.

TIPO III — Ferro-Cromo (FeCr)

As fitas de Ferro-Cromo ou Ferricromo são um pouquinho melhores do que as do Cromo, no que diz respeito à gravação de programas, mas sua raridade e alto custo não compensam seu uso. Só há fitas Ferro-Cromo importadas e são compatíveis com gravadores que aceitem fitas de Cromo (que por sua vez são compatíveis com as fitas de Óxido de Ferro).

TIPO IV — Metal (Ferro puro)

As fitas de Ferro puro são as melhores já fabricadas. Diferentes das outras, compostas por fita plástica impregnada com partículas metálicas, as fitas deste tipo são feitas exclusivamente por uma fina lâmina de metal puro (Metal Tape, como também são conhecidas). São caríssimas e raras no Brasil (por enquanto) e há poucos fabricantes no mundo, todos estrangeiros. Fitas assim exigem gravadores apropriados, com tecnologia "Metal", e geralmente aceitam qualquer tipo de fita.

O uso de uma fita incompatível com um gravador fatalmente acarretará seu excessivo desgaste, sua magnetização precoce e fitas impropriamente gravadas.

Duração

Outro critério importante na escolha da fita apropriada é sua duração (tempo da fita, não sua durabilidade). Fitas de longa duração (long play) do tipo 180 minutos, 120 minutos e 90 minutos são impróprias para serem usadas com programas porque, para caber mais fita no mesmo espaço, os fabricantes são forçados a fazerem suas fitas mais finas, em detrimento de sua qualidade. Em fitas assim, há menos partículas metálicas por extensão de fita e acontece um fenômeno que chamo "papel-carbono", e tira a nitidez das informações gravadas. O efeito "papel-carbono" acontece quando, estando a fita enrolada e com isto as partículas de um trecho ficando superpostas às de outro trecho em outra camada e sendo a película plástica bem fina, as informações de uma camada "migram" para outra, perdendo assim a nitidez da informação gravada, num fenômeno semelhante à tinta que passa de uma folha a outra em folhas finas de um caderno.

No que diz respeito a gravadores, não se pode estabelecer preferências sem ter comprovado seu uso pessoalmente ou por intermédio de alguém que o possua e esteja satisfeito com seu uso. Vale lembrar que os mais caros nem sempre são os melhores.

No próximo número darei mais dicas e ensinarei como identificar e sanar as possíveis malfunções do gravador de maneira "caseira".

Até lá.

GRAVAÇÃO DE PROGRAMAS

Nancy Mitie Ariga

Muitos dos nossos leitores escrevem-nos falando sobre seus problemas com gravação de programas. Você provavelmente já teve um problema semelhante a este:

"Depois de ter digitado durante algumas horas um super-programa, resolvi gravá-lo. Após alguns dias, quando fui carregá-lo, esperei vários minutos para que o programa fosse carregado (era um programa com quase 1Ø kBytes). Aparentemente parecia-me que estava tudo bem. Mas no final . . . O PROGRAMA NÃO HAVIA PASSADO . . . Tentei várias outras vezes e nada . . . "quase subi pelas paredes" . . . até que consegui encontrar o volume correto."

Problemas de gravação podem ser por três motivos:

1) Má gravação;
2) Cabeçote não regulado;
3) Volume alto ou baixo demais.

No primeiro caso, a má gravação pode ter sido causada pelo uso de fitas de má qualidade, volume muito baixo durante a gravação, cabeçote sujo causando flutuações na gravação e inúmeras outras razões. Neste caso, sinto muito, mas você perdeu o seu programa e terá que digitá-lo novamente.

Para o segundo caso, leia a seção HOBBY deste número, onde nosso colaborador Tanios Hamzo explica como conservar o seu cabeçote limpo e como regulá-lo.

A dica deste mês refere-se ao terceiro caso, ou seja, ajudará você a verificar se o volume está correto ou não. Para isso, antes de digitar um programa-monstruoso, grave antes um programa-teste, por exemplo, este:

```
   1 PRINT "SEU PROGRAMA-TESTE F
OI CARREGADOPORTANTO O VOLUME ES
TA" CORRETO  VOU AGORA CARREGAR
O PROGRAMA"
   2 PAUSE 300
   3 LOAD ""
   4 SAVE "PROGRAMA-TESTE"
   5 RUN
```

Grave-o sempre no começo da fita através da instrução GOTO 4. Assim, quando o programa-teste terminar de ser gravado pressione a tecla BREAK para que o programa pare e digite NEW.

Digite o seu programa e depois grave-o com o mesmo volume do programa-teste e sua fita ficará assim:

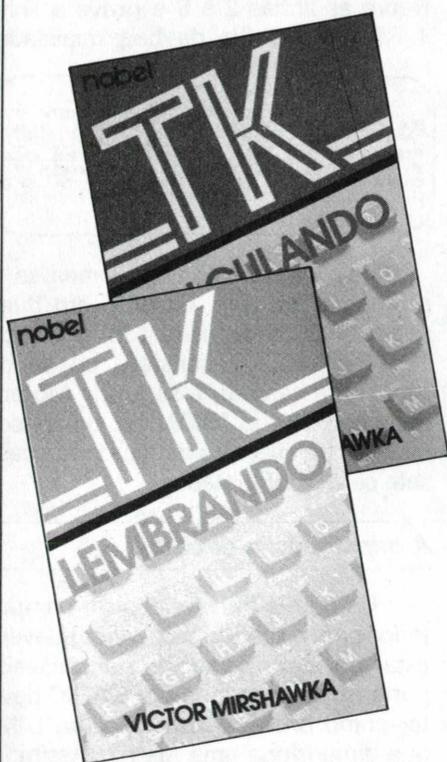
PROGRAMA-TESTE	GAP	PROGRAMA

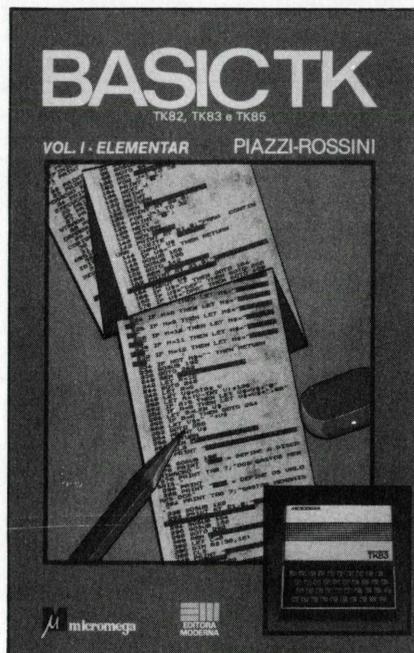
Portanto, para carregar o seu programa, primeiramente será carregado o programa-teste. Se o programa-teste não "passar", o problema pode ser do volume (alto ou baixo demais). Tente regulá-lo (verifique antes se a gravação não possui flutuações enquadrando-se no primeiro caso).

Se o programa-teste for carregado, aparecerá na tela:

```
SEU PROGRAMA-TESTE FOI CARREGADO
PORTANTO O VOLUME ESTA" CORRETO
 VOU AGORA CARREGAR O PROGRAMA
```

e depois de cinco segundos, o computador começa a carregar o programa (veja a linha 3 do programa-teste) e depois de alguns minutos o seu programa estará na memória do seu computador. Experimente fazer um teste e diga adeus aos problemas de gravação! . . . (pelo menos problemas com volume). ○

LIVROS

Editora: Livraria Nobel — Editora-Distribuidora
Autor: Victor Mirshawka

Estes três livros da série da Nobel são especialmente dedicados para os micros fabricados pela Microdigital. Destinados aos iniciantes em computação, os livros se propõem a fornecer as informações básicas para quem está pegando nos TK's pela primeira vez. A coleção, segundo a editora, compreende um total de 100 programas que possibilitam ao iniciante, a programação de seus próprios jogos, a resolução dos problemas de Matemática, Física, Química e de Finanças e Engenharia. *A.L.A.*

Editora: Micromega/PMD Ltda.
Autores: Flávio Rossini, Pierluigi Piazzi

Um livro de BASIC é algo tão comum no mercado que não mereceria destaque, a não ser que tivesse algo diferente. Este é o caso do livro BASIC TK.

Voltado para principiantes, complementa o manual dos fabricantes de computadores compatíveis com o TK, com uma linguagem simples e direta, cobrindo as principais dúvidas que podem ser geradas no trato com a máquina.

O BASIC-TK foi baseado integralmente no Curso de BASIC publicado nas páginas de Microhobby, a partir do número 1 (aí está uma excelente oportunidade aos leitores que passaram a conhecer nossa revista recentemente e não tiveram oportunidade de conhecer as primeiras lições do nosso curso), com uma didática excelente, não deixando o leitor na mão em nenhum de seus capítulos. *A.D.*

FUNÇÕES DE ARMAZENAMENTO PARA O TK-83

Funções de Armazenamento
Memória Ocupada programa 2 = 96 bytes

Nível =

Computadores = TK-83 e compatíveis

Na revista número 5 da Microhobby, falamos sobre as funções de armazenamento de dados do TK-85.

DSAVE, DLOAD e DVERIFY são sub-rotinas da ROM do TK-85 que permitem o armazenamento de dados numa fita cassete, ou seja, somente os dados são gravados na fita. Mas estas sub-rotinas não estão na ROM do TK-82 e 83. Por isso fizemos um programa em linguagem de máquina que tem a mesma função do DSAVE e do DLOAD que são funções que gravam e recuperam dados de uma fita cassete.

Os Programas

Primeiro digite o programa 1 para a introdução do programa em linguagem de máquina na linha 1 REM. A linha 1 REM deve ter 90 pontos (Sugestão: utilize o programa "A linha REM" da revista número 7 da Microhobby).

```
   1 REM ........................
................................
................................
...
   2 FOR I=16514 TO 16603
   3 SCROLL
   4 PRINT I;"..";
   5 INPUT A$
   6 LET N=16*CODE A$+CODE A$(2)
-476
   7 POKE I,N
   8 PRINT A$
   9 NEXT I
```

Digite RUN e introduza os códigos em hexadecimal. Apresentamos esta listagem de uma forma diferente: colocamos também os mnemônicos (descrição abreviada dos códigos em linguagem de máquina). Isso permitirá a quem conhecer um pouco de Assembly, observar melhor o funcionamento do programa. Mas lembre-se que para cada endereço você deve digitar um código de dois dígitos. Por exemplo, no endereço 16514 digite "CD", no endereço 16515 "E7", e assim por diante.

```
LISTAGEM HEXADECIMAL

16514..CD  CALL   743
16515,.E7
16516..02
16517..06  LD     B,14
16518..0E
16519..21  LD     HL,65280
16520..00
16521..FF
16522..2B  DEC    HL
16523..7C  LD     A,H
16524..B5  OR     L
16525..20  JR     NZ,-5
16526..FB
16527..10  DJNZ   -10
16528..F6
16529..2A  LD     HL,(16400)
16530..10
16531..40
16532..23  INC    HL
16533..4E  LD     C,(HL)
16534..23  INC    HL
16535..46  LD     B,(HL)
16536..23  INC    HL
16537..C5  PUSH   BC
16538..5E  LD     E,(HL)
16539..CD  CALL   799
16540..1F
16541..03
16542..C1  POP    BC
16543..0B  DEC    BC
16544..78  LD     A,B
16545..B1  OR     C
16546..20  JR     NZ,-12
16547..F4
16548..C9  RET
16549..CD  CALL   743
16550..E7
16551..02
16552..2A  LD     HL,(16400)
16553..10
16554..40
16555..23  INC    HL
16556..4E  LD     C,(HL)
16557..23  INC    HL
16558..46  LD     B,(HL)
16559..23  INC    HL
16560..C5  PUSH   BC
16561..1E  LD     E,8
16562..08
16563..DB  IN     A,(254)
16564..FE
16565..D3  OUT    (255),A
16566..FF
16567..17  RLA
16568..30  JR     NC,-7
16569..F9
16570..0E  LD     C,148
16571..94
16572..06  LD     B,26
16573..1A
16574..0D  DEC    C
16575..DB  IN     A,(254)
16576..FE
16577..17  RLA
16578..CB  BIT    7,C
16579..79
16580..79  LD     A,C
16581..38  JR     C,-11
16582..F5
16583..10  DJNZ   -4
16584..F5
16585..20  JR     NZ,4
16586..04
16587..FE  CP     86
16588..56
16589..30  JR     NC,-28
16590..E4
16591..3F  CCF
16592..CB  RL     (HL)
16593..16
16594..1D  DEC    E
16595..20  JR     NZ,-34
16596..DE
16597..C1  POP    BC
16598..0B  DEC    BC
16599..78  LD     A,B
16600..B1  OR     C
16601..C8  RET    Z
16602..18  JR     -45
16603..D3
```

Ao terminar, digite o comando LIST e veja como sua linha REM ficou. Retire as linhas 2 a 8 e grave a linha 1 REM numa fita de boa qualidade.

```
   1 REM LN SCROLL ▚:5  COPY F?
▛4 CLS ( PLOT E(RND7??7??VAL ?LN
3▀AT "?▛4 POKE TAN LN SCROLL ▘E!
RND7??7??VAL 2▚(= RETURN PEEK COP
Y *K RAND :▛▘,$(= RETURN *ACS ??
5 PRINT ( PRINT 4▖ RETURN ?K SLO
W ZACS -14 THEN AT "?▛COS /PEEK
```

Veja os mnemônicos e analise o programa: ele está dividido em duas sub-rotinas: a primeira (do endereço 16514 ao 16548) transfere os dados para a fita cassete; a segunda (do endereço 16549 ao 16603) faz o inverso, ou seja, transfere os dados da fita cassete para o computador.

A transferência de dados

Os dados transferidos do computador para a fita (ou vice-versa) devem estar no início da região de variáveis, portanto, o programa em BASIC deve ter como primeira instrução um DIM, que dimensiona uma matriz. Assim, o dimensionamento de uma matriz deverá ser a primeira coisa a ser executada, reservando assim, uma área para colocar os dados. Portanto, toda vez que você for utilizar estas sub-rotinas, você deve em primeiro lugar carregar este programa e depois elaborar um programa BASIC para a manipulação dos dados, utilizando o comando USR para acionar as sub-rotinas em linguagem de máquina.

Um exemplo: Um arquivo-dicionário

Para exemplificar o funcionamento destas sub-rotinas, vamos criar um arquivo-dicionário, com palavras em português e inglês. Assim, quando você quiser saber como uma determinada

palavra em português é escrita em inglês, ou vice-versa, poderá utilizar este arquivo que será guardado numa fita cassete.

SAVE-dados

Carregue o programa 2 e depois digite as linhas 10 a 200 do programa 3.

```
   1 REM LN SCROLL █:5  COPY F?
█4 CLS ( PLOT E(RND7???7VAL ?LN
3█AT "?█4 POKE TAN LN SCROLL █E(
RND7???7VAL 2█<= RETURN PEEK COP
Y *K RAND :█,$<= RETURN *ACS ??
S PRINT ( PRINT 4█ RETURN ?K SLO
W ZACS -14 THEN AT "?█COS /PEEK
  10 DIM A$(10,20)
  20 DIM P$(10)
  30 DIM I$(10)
  40 FOR I=1 TO 10
  50 SCROLL
  60 PRINT I;"..";
  70 INPUT P$
  80 PRINT P$;"..";
  90 INPUT I$
 100 PRINT I$
 110 LET A$(I)=P$+I$
 120 NEXT I
 130 CLS
 140 PRINT "P/GRAVAR PLAY REC E
NL."
 150 INPUT P$
 160 LET X=USR 16514
 170 SLOW
 180 STOP
 190 SAVE "D-SAV█"
 200 RUN
```

Como você pode notar, o dimensionamento da matriz A$ é a primeira instrução a ser executada.

Ponha o programa em funcionamento e digite palavras em português e em inglês, num total de 10 palavras. Por exemplo: digite "MEMÓRIA"

```
1..MEMORIA..
"█"
```

e depois MEMORY

```
1..MEMORIA..MEMORY
2..
"█"
```

Continue digitando até completar o arquivo e o programa avisar que está pronto para gravar. Acione o REC do seu gravador, verifique se os MICs estão conectados e NEW LINE. O comando USR 16514 é acionado, ou seja, a sub-rotina SAVE-dados, transferirá os dados da matriz A$ para a fita. Quando ele terminar de gravar os dados, o programa pára. Se você desejar aumentar o seu arquivo, você precisa modificar a dimensão de A$, pois A$ (10, 20) significa que teremos 10 palavras em inglês/português, tendo 20 dígitos cada, onde as 10 primeiras são da palavra em português e as outras 10 da palavra em inglês.

Por exemplo:

A$(1)="MEMORIAbbbMEMORYbbbb"

Modifique também a linha 20 e 30 pois P$ e I$ será a palavra em português e inglês introduzidas por você e a linha 40.

Para gravar o programa e os dados (caso você não tenha dado CLEAR), digite GOTO 190.

LOAD-dados

Vamos agora ver o funcionamento desta sub-rotina. Carregue o programa 2 e digite as linhas 10 a 610 do programa 4.

```
   1 REM LN SCROLL █:5  COPY F?
█4 CLS ( PLOT E(RND7???7VAL ?LN
3█AT "?█4 POKE TAN LN SCROLL █E(
RND7???7VAL █<= RETURN PEEK COP
Y *K RAND :█,$<= RETURN *ACS ??
S PRINT ( PRINT 4█ RETURN ?K SLO
W ZACS -14 THEN AT "?█COS /PEEK
  10 DIM B$(10,20)
  15 DIM C$(10)
  20 PRINT "PALAVRA PROCURADA?"
  30 INPUT C$
  40 CLS
  50 PRINT AT 1,5;C$;AT 3,3;"E█U
MA PALAVRA EM:";AT 5,5;"1. INGLE
S";TAB 5;"2. PORTUGUES"
  60 INPUT M
  70 GOTO M*100
 100 REM INGLES
 110 GOSUB 500
 120 FOR I=1 TO 10
 130 LET A$=B$(I,1 TO 10)
 140 IF C$=B$(I,11 TO 20) THEN G
OTO 400
 150 NEXT I
 160 GOTO 300
 200 REM PORTUGUES
 210 GOSUB 500
 220 FOR I=1 TO 10
 230 LET A$=B$(I,11 TO 20)
 240 IF C$=B$(I,1 TO 10) THEN GO
TO 400
 250 NEXT I
 300 CLS
 310 PRINT "NAO ACHEI A PALAVRA
NO ARQUIVO"
 320 PRINT AT 2,4;"1. PROCURAR E
M OUTRO ARQUIVO"
 330 PRINT TAB 4;"2. PARAR"
 340 INPUT N
 350 IF N=1 THEN GOTO 70
 360 IF N=2 THEN STOP
 400 CLS
 410 PRINT C$;A$
 420 STOP
 500 REM D-LOAD
 510 CLS
 520 LET X=USR 16549
 530 SLOW
 540 PRINT "DESLIGUE O GRAVADOR"
 550 FOR I=1 TO 60
 560 NEXT I
 570 RETURN
 600 SAVE "D-LOA█"
 610 RUN
```

A matriz onde serão colocados os dados da fita não precisa ter o mesmo "label" (nome) que a matriz gravada na fita, mas apenas a mesma dimensão da matriz A$ e ser a primeira instrução a ser executada em BASIC.

Dê RUN e digite a palavra que você deseja saber o significado. Por exemplo, digite "BOOK" e na tela aparecerá:

```
BOOK
E█UMA PALAVRA EM:
  1. INGLES
  2. PORTUGUES
```

Digite "1" pois BOOK é uma palavra em inglês, abaixe um pouco o volume do gravador, acione o PLAY do gravador, verifique se os EARs estão conectados e NEW LINE.

A sub-rotina LOAD-dados é acionada através do comando USR 16549, transferindo para a matriz B$ os dados da fita. Ao terminar a transferência de dados, uma mensagem pedindo para desligar o computador aparecerá na tela. Inicia-se então a comparação entre a palavra que você deseja saber e as palavras do arquivo. Se a palavra for encontrada, aparecerá na tela a palavra e a sua tradução.

```
BOOK      LIVRO
```

Caso contrário:

```
NAO ACHEI A PALAVRA NO ARQUIVO
VOCE DESEJA:
  1. PROCURAR EM OUTRO ARQUIVO
  2. PARAR
```

Pressione a tecla "1" para carregar outro arquivo se ele existir ou "2" se você não possui outro arquivo, e deseja parar.

Para gravar este programa digite GOTO 600 e para a procura de uma outra palavra, digite RUN.

Agora que você já sabe como utilizar as sub-rotinas do programa 2, crie outros arquivos e boa sorte!

○

Coordenação:
Prof. Wilson José Tucci/Domus Informática
Colaborações de
Luiz C. N. Szente
José Eduardo Moreira
Daniel R. Falconer

Introdução Geral

Com a chegada ao mercado do novo computador fabricado pela Microdigital, a equipe que produz a coluna "POR DENTRO DO APPLE" decidiu abrir uma nova coluna específica para o TK-2000.

O TK-2000 é muito semelhante ao APPLE, e a maioria dos programas que foram escritos para o APPLE funcionam sem problemas no TK-2000. No entanto, como existem algumas diferenças entre os dois, nasceu a idéia de escrevermos artigos sobre este assunto.

Esta coluna tem por objetivo explorar o hardware e o software do TK-2000 através de programas, utilizando todo o seu potencial. Esperamos que os leitores nos enviem sugestões e dúvidas, para que assim possamos adequar nossa matéria à um número maior de pessoas.

Assim, vamos à luta.

Muitos leitores, escreveram solicitando mais artigos sobre animação e parte gráfica. Faremos então, uma série de artigos enfocando este assunto com ênfase nas características do TK-2000.

Translação animada de figuras (em 2 dimensões)

```
10  HGR
20  HCOLOR= 7
30  REM  COLOCACAO DO DESENHO
40  DIM XI(20),YI(20),XF(20),YF(20)
50  DIM X1(20),Y1(20),X2(20),Y2(20)
60 NP = 1
70  PRINT "COORD. DOS VERT. P/ FIM TECLE NUM. > 300"
80  INPUT "XI,YI,XF,YF ";A,B,C,D
90  IF A > 300 OR B > 300 OR C > 300 OR D > 300 THEN  GOTO 250
100  HPLOT A,B TO C,D
110  HOME
120  PRINT "LINHA CERTA ??? (S/N) ";
130  GET A$
140  IF A$ = "S" THEN  GOTO 190
150  HCOLOR= 0
160  HPLOT A,B TO C,D
170  HCOLOR= 7
180  GOTO 70
190 XI(NP) = A
200 YI(NP) = B
210 XF(NP) = C
220 YF(NP) = D
230 NP = NP + 1
240  GOTO 70
250  HOME
260  PRINT "COLOQUE ORIGEM E FIM DA TRANSL. "
270  INPUT "X0,Y0,X,Y ";X0,Y0,X,Y
280  HOME : HGR
290  INPUT "NUMERO DE FIGURAS ";NF
300 XL = X - X0
310 YL = Y - Y0
320 D =  SQR (XL ^ 2 + YL ^ 2)
330 DX = XL / D
340 DY = YL / D
350 DD = D / (NF - 1)
360  FOR J = 1 TO NF
370 XX = X0 + DX * DD * (J - 1)
380 YY = Y0 + DY * DD * (J - 1)
390  FOR I = 1 TO NP
400 X1(I) = XX + XI(I)
410 Y1(I) = YY + YI(I)
420 X2(I) = XX + XF(I)
430 Y2(I) = YY + YF(I)
440  NEXT I
450  FOR K = 1 TO NP
460  HPLOT X1(K),Y1(K) TO X2(K),Y2(K)
470  NEXT K
480  NEXT J
490  END
```

Usaremos a técnica de se guardar em vetores as coordenadas das partes dos vértices que formam a figura.

Para se transladar uma figura, é necessário fornecer a coordenada inicial e final do movimento. O programa deve desenhar a figura nos pontos intermediários e, a cada impressão, apagar a precedente para se criar a ilusão de movimento.

Uma vez escrito o programa, execute-o e coloque os seguintes pontos (digite S ao surgir a mensagem "LINHA CERTA? (S/N)"):

```
10,3,15,3
15,3,25,10
25,10,55,10
55,10,70,15
70,15,10,15
10,15,10,3
40,12,25,20
25,20,15,20
15,20,30,12
30,12,40,12
```

Você deve ter formado uma imagem estilizada de um avião.

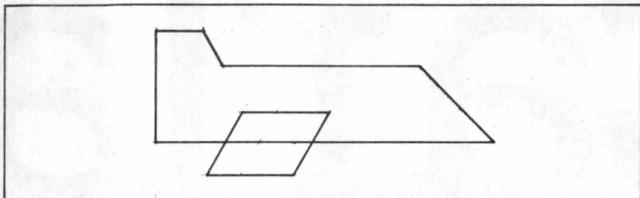



Fig. 1

Digite agora:

999,999,999

Coloque agora os pontos da movimentação.

Ex.: 10, 10, 150, 150

coque o número de figuras intermediárias.

Ex.: 8

Deverá ter surgido na tela a figura se movimentando da coordenada 10, 10 à coordenada 150, 150, deixando traçado na tela as figuras intermediárias.

Para "apagá-las", a fim de que se de a ilusão de movimento, coloque as seguintes linhas:

```
55  DIM X3(20),Y3(20),X4(20),Y4(20)
392 X3(I) = X1(I)
394 Y3(I) = Y1(I)
396 X4(I) = X2(I)
398 Y4(I) = Y2(I)
442  HCOLOR= 0
444  FOR K = 1 TO NP
446  HPLOT X3(K),Y3(K) TO X4(K),Y4(K)
448  NEXT
449  HCOLOR= 7
```

As linhas 10 à 50 representam a inicialização das variáveis.

Nos vetores XI e YI serão colocadas as coordenadas iniciais do traçado de uma linha e nos vetores XF e YF as finais.

Nos vetores X1, Y1 e X2, Y2, serão colocadas as coordenadas de movimentação.

Na variável NP, será armazenado o número de pontes da figura.

A entrada de dados é feita da linha 70 à 240. Devem ser fornecidas as coordenadas do vértice de uma linha. Ao se digitar um número maior que 300, o processo de entrada de dados finaliza. A cada entrada de dados o programa traça a linha e pergunta se a linha está correta. Caso não tenha ficado como se queira, o usuário tem a opção de "apagar" a linha e traçar outra.

Para se transladar uma figura, devem ser fornecidas a coordenada inicial e final da translação, bem como o número de imagens que se deseja para que se de a idéia de movimento.

Isto é feito das linhas 260 à 290.

Das linhas 300 à 350, estão as considerações matemáticas da movimentçaão (figura 2):

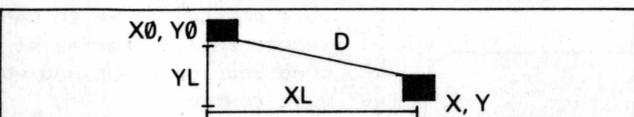



Fig. 2

XL é a distância entre as partes inicial e final medida no eixo X.

YL é a distância entre as partes inicial e final medida no eixo Y.

D é a distância entre os pontos propriamente dita. Sobre esse eixo é que se fará a translação.

DX representa o cosseno do ângulo formado pelas letras D e XL.

DY representa o cosseno do ângulo formado pelas letras D e YL.

DD representa os intervalos onde serão colocadas as figuras intermediárias.

Das linhas 360 à 480 são calculadas as coordenadas dos pontos das figuras a serem traçadas.

XX representa o incremento a ser dado no eixo X. É calculado, somando-se à coordenada inicial de X, o valor da distância multiplicado pelo cosseno do ângulo multiplicado pelo número de figuras que vai se traçar. Análogo para YY.

A linha 55 dimensiona os vetores que armazenaram as coordenadas "anteriores" (que irão apagar).

A "figura anterior" será "apagada" nas linhas 442 à 448.

Esperamos ter ensinado uma maneira de se dar animação às figuras. Faça várias delas e não tenha medo de errar. A qualquer tempo, o programa apaga a letra errada. Nos próximos números daremos mais dicas sobre animação.

Boa sorte e bons desenhos.

OBS.: Em alguns equipamentos diferentes do TK-2000, sugerimos alterar as linhas 110 e 250 para HOME: VTAB21.

Todas as listagens foram tiradas numa impressora Mônica da Elebra.

Carla

Que tal uma conversa com uma psicanalista num sábado à noite, na sua casa? Seria interessante, não? . . . Mais ainda se "Carla" nada mais fosse do que seu querido TK.

Este programa tem a finalidade de simular inteligência. As respostas que "ela" dá muitas vezes parecem vir de um ser humano (aparentemente normal . . .) e podem tapear os usuários mais incautos. Ele foi desenvolvido em homenagem a simpática (e linda!!!) prima do Sr. Nabor Rosenthal, que é psicanalista em Penêdo e chama-se Carla também. Ela gostou do programa porque testava a reação em seus pacientes com relação a aceitação de computadores como amigos, além de verificar o teste de Turing (veja o artigo na revista nº 8 — "Podem as máquinas pensar?").

— Oi. Eu sou Carla, conte-me seu problema.
— Os outros me perseguem.
— Você pode desenvolver este assunto?
— Eu sou seguido.
— Há quanto tempo você é seguido?
— Desde que minha mãe me abandonou.
— Não falemos sobre mães. Não sou freudiana.
— Você também me persegue.
— Por algum acaso você está falando de mim?
— Eu sinto que você me odeia.
— Conte-me mais a respeito do que você sente.
— Tenho mêdo de computador.
— Computadores te preocupam?
— Só os do seu tipo.
— Porque você está preocupado com meu tipo?
(Diálogo entre Carla e o pessoal da Redação)

Observação: Nas tabelas 3 e 4, dê sempre um espaço antes ou depois de cada palavra. O acento agudo é obtido por GRAFICS SHIFT 1.

TABELA 1

1	1	3
2	4	2
3	6	4
4	6	4
5	25	3
6	14	3
7	17	3
8	20	2
9	22	3
10	18	4
11	26	4
12	28	4
13	32	3
14	35	5
15	40	9
16	40	9
17	40	9
18	40	9
19	40	9
20	40	9
21	49	2
22	51	4
23	55	4
24	59	4
25	63	1
26	63	1
27	64	5
28	69	5
29	74	2
30	76	4
31	80	3
32	83	7
33	90	3
34	93	6
35	99	7
36	116	3
37	113	3
38	119	2
39	121	2
40	106	7

TABELA 2

1	VOCE PODE
2	EU POSSO
3	VOCE E'
4	VOCE E'
5	EU NAO POSSO
6	EU SINTO
7	PORQUE VOCE NAO
8	PORQUE EU NAO POSSO
9	VOCE E'
10	EU NAO
11	EU SOU
12	SOU
13	VOCE
14	EU QUERO
15	O QUE
16	COMO
17	QUEM
18	AONDE
19	QUANDO
20	POR QUE
21	NOME
22	PORQUE
23	DESCULPE
24	SONHO
25	COMO VAI
26	OI
27	TALVEZ
28	NAO
29	SEU
30	SEMPRE
31	PENSO
32	PARECE
33	SIM
34	AMIGO
35	COMPUTADOR
36	CARLA
37	DINHEIRO
38	MAE
39	SUICIDIO
40	INFOR. NAO ACHADA

TABELA 3

1	E'
2	FOI
3	VOCE
4	SEU
5	EU TENHO
6	EU SOU
7	VOCE

TABELA 4

1	SOU
2	FUI
3	EU
4	MEU
5	VOCE TEM
6	VOCE E'
7	MIM

TABELA 5

```
1   VOCE NAO ACREDITA QUE EU POS
SA*
2   TALVEZ VOCE GOSTASSE DE SER
CAPAZ DE*
3   VOCE QUER QUE EU SEJA CAPAZ
DE*
4   TALVEZ VOCE NAO QUEIRA*

5   VOCE QUER SER CAPAZ DE*

6   O QUE FAZ VOCE PENSAR QUE EU
 SOU *
7   VOCE FICA FELIZ ACREDITANDO
QUE EU SOU*
8   TALVEZ VOCE GOSTASSE DE SER*

9   ALGUMAS VEZES VOCE GOSTARIA
DE  SER*
10  REALMENTE VOCE NAO *

11  POR QUE VOCE NAO*

12  POR QUE VOCE DESEJA SER CAPA
Z   DE*
13  ISTO PREOCUPA VOCE ?

14  CONTE-ME MAIS A RESPEITO DO
QUE VOCE SENTE.
15  VOCE FREQUENTEMENTE SENTE*

16  VOCE GOSTA DE SENTIR *

17  REALMENTE VOCE NAO ACREDITA
*
18  TALVEZ EM UM BOM TEMPO EU QU
EIRA*
19  VOCE QUER QUE EU *

20  VOCE ACHA QUE VOCE DEVERIA S
ER  CAPAZ DE *
21  POR QUE VOCE NAO PODE *

22  POR QUE VOCE ESTA INTERESSAD
O SE EU SOU OU NAO *
23  VOCE PREFERIRIA SE EU NAO FO
SSE*
24  TALVEZ EM SUAS FANTASIAS EU
SEJA *
25  COMO VOCE SABE QUE NAO PODE
*
26  VOCE TENTOU ?

27  TALVEZ AGORA VOCE POSSA *

28  POR ACASO VOCE VEIO A MIM PO
R   QUE VOCE E' *
29  A QUANTO TEMPO VOCE E' *

30  VOCE ACREDITA QUE E' NORMAL
SER *
```

Como Carla simula inteligência?

Através do programa, "ela" procura palavras-chaves na resposta do usuário usando várias tabelas (matrizes) que "escolhem" a melhor resposta pronta e ainda soma a esta um "pedaço" significativo da resposta do usuário. A tarefa de "procura" das palavras se dá na sub-rotina da linha 1220 que é uma conversão da função INSTR (D$, W$), existente em alguns BASICs, para a linguagem do TK. Ela tem por função localizar a string W$ em D$ e "dizer" onde ela começa. Por exemplo: Se você tem a palavra "HEI" estando em W$ e D$ for "dinheiro" o valor da função INSTR (D$, W$) será 4. Se caso não achar W$ em D$ ela valerá zero. INSTR é abreviação de IN STRING que quer dizer "na string".

Existem 5 tabelas:

1ª TABELA — consta de três números: o primeiro corresponde ao número da palavra-chave na segunda tabela, o segundo diz a primeira resposta a ser dada e o terceiro, quantas respostas existem para aquela palavra.

2ª TABELA — palavras-chave principais.

3ª e 4ª TABELAS — são palavras que devem ser trocadas entre si para dar certa concordância na frase resposta de Carla.

5ª TABELA — respostas prontas ou semi-prontas.

Basicamente uma resposta coerente depende do uso de palavras-chave. Se o usuário conhece estas palavras, pode haver um diálogo inteligente e não repetitivo. Se não conhecer pode ocorrer coisas estranhas e algumas respostas absurdas. Dá para se divertir bastante com as duas alternativas.

Manipulação do programa

Após ter digitado o programa principal deve-se dar GOTO 10. Serão introduzidas todas as tabelas (na 3ª e 4ª deixe sempre um espaço no começo e no fim de cada "string") e o programa parará. Sugiro gravá-lo (várias vezes) por segurança, porque é muito extenso. Para rodá-lo dê GOTO 260 e ***nunca, nunca mesmo*** dê RUN ou todas as tabelas serão apagadas (não nos responsabilizamos por "cabelos arrancados" em caso de RUN!!!). Se caso você quiser ler as tabelas dê GOTO 850 e CONT para continuar a leitura, quando a tela estiver cheia.

```
31   VOCE SE DIVERTE SENDO *
32   ESTAMOS DISCUTINDO VOCE--NAO
 EU.
33   OH,EU *
34   POR ALGUM ACASO VOCE ESTA* FA
LAN-DO DE MIM ?
35   O QUE SIGNIFICARIA A VOCE SE
 EU *
36   POR QUE VOCE QUER *
37   SUPONHA QUE VOCE LOGO CONSIG
A *
38   O QUE SIGNIFICARIA A VOCE SE
     VOCE *
39   AS VEZES EU TAMBEM QUERO *
40   POR QUE VOCE PERGUNTA ?
41   ESTA QUESTAO INTERESSA A VOC
E ?
42   QUE RESPOSTA VOCE GOSTARIA ?
43   O QUE VOCE ACHA ?
44   VOCE SEMPRE TEM TANTAS PERGU
NTAS EM MENTE ?
45   O QUE E* QUE VOCE REALMENTE
QUER SABER
46   VOCE FEZ ESTA PERGUNTA A MAI
S    ALGUEM ?
47   VOCE JA FEZ TAIS PERGUNTAS
     ANTES ?
48   O QUE MAIS LHE A MENTE QUAND
O    ME PERGUNTA ISSO ?
49   NOMES NAO ME INTERESSAM.
50   NAO ME IMPORTO COM NOMES--PO
R    FAVOR CONTINUE.
51   ESTA E* A VERDADEIRA RAZAO ?
52   NENHUMA OUTRA RAZAO VEM A ME
NTE?
53   ESTA RAZAO EXPLICA QUALQUER
OU-  TRA COISA ?
54   QUAIS OUTRAS RAZOES PODEM EX
IS-  TIR ?
55   POR FAVOR,NAO SE DESCULPE.
56   DESCULPAS NAO SAO NECESSARIA
S.
57   QUE SENTIMENTOS VOCE TEM QUA
NDO PEDE DESCULPAS ?
58   NAO SEJA TAO DEFENSIVO.
59   O QUE ESTE SONHO TE SUGERE ?
60   VOCE SONHA COM FREQUENCIA ?
61   QUAIS PESSOAS APARECEM NOS S
EUS SONHOS ?
62   VOCE FICA PERTURBADO COM SEU
S    SONHOS ?
63   COMO VAI VOCE..POR FAVOR EXP
ONHA SEUS PROBLEMAS.
64   VOCE NAO PARECE TER MUITA CE
RTE-ZA.
65   POR QUE O TOM DE INCERTEZA ?
66   VOCE NAO PODE SER MAIS POSIT
IVO?
67   VOCE NAO TEM CERTEZA ?
68   VOCE NAO SABE ?
69   POR QUE NAO *
70   NAO DIGA NAO; E* SEMPRE TAO
NE-  GATIVO.
71   PORQUE NAO *
72   TEM CERTEZA ?
73   PORQUE NAO ?
74   POR QUE VOCE ESTA PREOCUPADO
 COM MEU *
75   POR QUE A RESPEITO DE VOCE M
ESMO *
76   VOCE NAO PODE PENSAR NUM EXE
MPLO ESPECIFICO ?
77   QUANDO ?
78   O QUE VOCE ESTA* PENSANDO ?
79   REALMENTE SEMPRE ?
80   VOCE REALMENTE PENSA ASSIM ?
81   MAS VOCE NAO TEM CERTEZA QUE
     VOCE *
82   VOCE DUVIDA QUE VOCE *
83   DE QUE MANEIRA ?
84   QUE SEMELHANCA VOCE VE ?
85   O QUE ESTA SEMELHANCA LHE SU
GE-  RE ?
86   QUE CONEXOES VOCE VE ?
87   PODERIA REALMENTE EXISTIR AL
GUMA CONEXAO ?
88   COMO ?
89   VOCE PARECE TOTALMENTE POSIT
IVO.
90   VOCE TEM CERTEZA ?
91   EU PERCEBO...
92   EU ENTENDO.
93   POR QUE VOCE TROUXE O ASSUNT
O    "AMIGO"?
94   OS SEUS AMIGOS TE PREOCUPAM
?
95   OS SEUS AMIGOS TE CHATEIAM ?
96   VOCE TEM CERTEZA QUE TEM ALG
UM   AMIGO ?
97   VOCE SE IMPOE AOS SEUS AMIGO
S ?
98   TALVEZ SEU AMOR PELOS AMIGOS
 O   DEIXE PREOCUPADO
99   COMPUTADORES TE PREOCUPAM ?
100 VOCE ESTA FALANDO A MEU RESP
EITO EM PARTICULAR?
101 VOCE TEM MEDO DE MAQUINAS ?
102 VOCE MENCIONOU COMPUTADORES
?
103 MAQUINAS TEM A VER COM SEUS
PRO-BLEMAS ?
104 V.NAO ACHA QUE UM COMPUTADOR
 PO-DE AJUDAR ALGUEM?
105 O QUE COM RELACAO A MAQUINAS
 TE PREOCUPA ?
106 DIGA-ME,VOCE TEM ALGUM PROBL
EMA PSICOLOGICO ?
107 O QUE ISTO LHE SUGERE ?
108 EU PERCEBO...
109 NAO TENHO CERTEZA SE ENTENDO
 VO-CE PERFEITAMENTE.
110 ADAPTE-SE A REALIDADE.QUAIS
OS   SEUS PENSAMENTOS?
111 VOCE PODE DESENVOLVER ESTE A
S-   SUNTO ?
112 ISTO E* DEVERAS INTERESSANTE
.
113 POR QUE VOCE TEM PROBLEMAS C
OM   DINHEIRO ?
114 VOCE ACHA QUE DINHEIRO E* TU
DO ?
115 VOCE TEM CERTEZA QUE O PROBL
EMA E* DINHEIRO ?
116 DEVEMOS CONVERSAR A SEU RESP
EI-  TO,NAO AO MEU.
117 O QUE A MEU RESPEITO ?
118 POR QUE VOCE SEMPRE TRAZ A T
ONA O MEU NOME?
119 NAO FALEMOS SOBRE MAES.NAO S
OU   FREUDIANA.
120 POR QUE ESSA OBSESSAO COM M
AES?
121 NAO PENSE NISSO.E* BESTEIRA.
122 CONVERSE MAIS COMIGO E MUDAR
A* DE IDEIA.
```

O Programa

```
  10 REM (C)-MICROMEGA 1983
  20 REM
  30 REM   P/ COMECAR: GOTO 260
  40 DIM S(40)
  50 DIM V(40)
  60 DIM K$(40,21)
  70 DIM A$(122,52)
  80 DIM T$(7,12)
  90 DIM U$(7,12)
 100 DIM R(40)
 110 DIM N(40)
 120 FOR X=1 TO 40
 130 CLS
 140 PRINT "(TABELA 1)ENTRE O 2"
,DEPOIS O 3""
 150 PRINT
 160 PRINT X;
 170 INPUT S(X)
 180 PRINT " ";S(X);" ";
 190 INPUT L
 200 PRINT L
 210 LET R(X)=S(X)
 220 LET N(X)=S(X)+L-1
 230 NEXT X
 240 CLS
 250 GOTO 1330
 260 SLOW
 270 PRINT "OI,EU SOU CARLA.","C
ONTE-ME SEU PROBLEMA"
 280 SLOW
 290 INPUT I$
 300 IF I$="" THEN GOTO 280
 310 FAST
 320 CLS
 330 LET I$=" "+I$+" "
 340 REM PROCURA DAS PALAVRAS DE
RECONHECIMENTO
 350 LET S=0
 360 FOR K=1 TO 40
 370 IF S>0 THEN GOTO 450
 380 LET W$=K$(K)( TO VAL K$(K)(
20 TO ))
 390 LET D$=I$
 400 GOSUB 1220
 410 IF NOT A THEN GOTO 450
 420 LET S=K
 430 LET T=A
 440 LET F$=W$
 450 NEXT K
 460 IF NOT S THEN GOTO 500
 470 LET K=S
 480 LET L=T
 490 GOTO 530
 500 LET K=40
 510 GOTO 710
 520 REM FASE DE CONJUGACOES
 530 LET C$=" "+I$(L+LEN F$ TO )
 540 FOR X=1 TO 7
 550 LET X$=T$(X)( TO VAL T$(X)(
11 TO ))
 560 LET Y$=U$(X)( TO VAL U$(X)(
11 TO ))
 570 FOR L=1 TO LEN C$
 580 IF L+LEN Y$>LEN C$ THEN GOT
O 630
 590 IF C$(L TO L+LEN Y$-1)<>Y$
THEN GOTO 630
 600 LET C$=C$( TO L-1)+X$+C$(L+
LEN Y$ TO )
 610 LET L=L+LEN X$
 620 GOTO 670
 630 IF L+LEN X$>LEN C$ THEN GOT
O 670
 640 IF C$(L TO L+LEN X$-1)<>X$
THEN GOTO 670
 650 IF X$=" VOCE " AND L>=LEN C
$-5 THEN GOTO 830
 660 LET C$=C$( TO L-1)+Y$+C$(L+
LEN X$-1 TO )
 670 NEXT L
 680 NEXT X
 690 IF C$(2)=" " THEN LET C$=C$
(2 TO)
 700 REM ACHA FRASE DE RESPOSTA
 710 LET F$=A$(R(K))( TO VAL A$(
R(K))(50 TO ))
 720 LET R(K)=R(K)+1
 730 IF R(K)>N(K) THEN LET R(K)=
S(K)
 740 PRINT "SUA RESPOSTA ANTERIO
R:"
 750 PRINT ">";I$
 760 PRINT AT 19,0;
 770 IF F$(LEN F$)="*" THEN GOTO
 810
```

```
780 PRINT F$
790 LET P$=I$
800 GOTO 280
810 PRINT F$( TO LEN F$-1),C$
820 GOTO 790
830 LET C$=C$( TO L-1)+" EU "
840 GOTO 690
850 PRINT "TABELA 1"
860 PRINT
870 FOR I=1 TO 40
880 PRINT I;TAB 4;S(I);TAB 8;N(
I)-S(I)+1
890 NEXT I
900 PRINT "TABELA 2"
910 PRINT
920 FOR I=1 TO 40
930 PRINT I;TAB 4;K$(I, TO 19)
940 NEXT I
950 PRINT "TABELA 3"
960 PRINT
970 FOR I=1 TO 7
980 PRINT I;TAB 4;T$(I, TO 10)
990 NEXT I
1000 PRINT
1010 PRINT "TABELA 4"
1020 PRINT
1030 FOR I=1 TO 7
1040 PRINT I;TAB 4;U$(I, TO 10)
1050 NEXT I
1060 PRINT
1070 PRINT "TABELA 5"
1080 PRINT
1090 FOR I=1 TO 122
1100 PRINT I;TAB 4;A$(I, TO 50)
1110 NEXT I
1120 STOP
1130 FOR N=1 TO 7
1140 CLS
1150 PRINT "(TABELA 3) PALAVRA-C
HAVE: ";N;" ?"
1160 INPUT I$
1170 LET T$(N)=I$
1180 LET T$(N)=T$(N)( TO 10)+STR
$ LEN I$
1190 PRINT T$(N)
1200 NEXT N
1210 RETURN
1220 REM A=INSTR(D$,U$)
1230 IF LEN U$=0 THEN GOTO 1310
1240 LET A=1
1250 LET Y=LEN U$
1260 IF Y>LEN D$ THEN GOTO 1310
1270 IF U$=D$(A TO Y) THEN RETUR
N
1280 LET A=A+1
1290 LET Y=Y+1
1300 GOTO 1260
1310 LET A=0
1320 RETURN
1330 REM PROGRAMA PARA CARREGAR
1340 CLS
1350 PRINT "ENTRE 40 PALAVRAS-CH
AVE"
1360 FOR N=1 TO 40
1370 CLS
1380 PRINT "(TABELA 2) PALAVRA-C
HAVE ";N;" ?"
1390 INPUT I$
1400 LET K$(N)=I$
1410 LET K$(N)=K$(N)( TO 19)+STR
$ LEN I$
1420 PRINT K$(N)
1430 NEXT N
1440 GOSUB 1130
1450 FOR N=1 TO 7
1460 CLS
1470 PRINT "(TABELA 4) PALAVRA-C
HAVE ";N;" ?"
1480 INPUT I$
1490 LET U$(N)=I$
1500 LET U$(N)=U$(N)( TO 10)+STR
$ LEN I$
1510 PRINT U$(N)
1520 NEXT N
1530 STOP
1540 FOR N=1 TO 122
1550 CLS
1560 PRINT "(TABELA 5) RESPOSTA
Nº ";N;":"
1570 PRINT
1580 INPUT I$
1590 PRINT I$
1600 INPUT P$
1610 IF P$<>"" THEN GOTO 1580
1620 CLS
1630 LET A$(N)=I$
1640 LET A$(N)=A$(N)( TO 50)+STR
$ LEN I$
1650 PRINT A$(N)
1660 NEXT N
1670 STOP
1680 SAVE "CARLA"
1690 PRINT "DIGITE BREAK"
1700 PAUSE 120
1710 GOTO 1680
1720 STOP
```

O Programa

```
  10 REM (C)-MICROMEGA 1983
  20 REM
  30 REM   P/ COMECAR: GOTO 260
  40 DIM S(40)
  50 DIM V(40)
  60 DIM K$(40,21)
  70 DIM A$(122,52)
  80 DIM T$(7,12)
  90 DIM U$(7,12)
 100 DIM R(40)
 110 DIM N(40)
 120 FOR X=1 TO 40
 130 CLS
 140 PRINT "(TABELA 1)ENTRE O 2ª
,DEPOIS O 3ª"
 150 PRINT
 160 PRINT X;
 170 INPUT S(X)
 180 PRINT " ";S(X);" ";
 190 INPUT L
 200 PRINT L
 210 LET R(X)=S(X)
 220 LET N(X)=S(X)+L-1
 230 NEXT X
 240 CLS
 250 GOTO 1330
 260 SLOW
 270 PRINT "OI,EU SOU CARLA.","C
ONTE-ME SEU PROBLEMA"
 280 SLOW
 290 INPUT I$
 300 IF I$="" THEN GOTO 280
 310 FAST
 320 CLS
 330 LET I$=" "+I$+" "
 340 REM PROCURA DAS PALAVRAS DE
 RECONHECIMENTO
 350 LET S=0
 360 FOR K=1 TO 40
 370 IF S>0 THEN GOTO 450
 380 LET W$=K$(K)( TO VAL K$(K)(
20 TO ))
 390 LET D$=I$
 400 GOSUB 1220
 410 IF NOT A THEN GOTO 450
 420 LET S=K
 430 LET T=A
 440 LET F$=W$
 450 NEXT K
 460 IF NOT S THEN GOTO 500
 470 LET K=S
 480 LET L=T
 490 GOTO 530
 500 LET K=40
 510 GOTO 710
 520 REM FASE DE CONJUGACOES
 530 LET C$=" "+I$(L+LEN F$ TO )
 540 FOR X=1 TO 7
 550 LET X$=T$(X)( TO VAL T$(X)(
11 TO ))
 560 LET Y$=U$(X)( TO VAL U$(X)(
11 TO ))
 570 FOR L=1 TO LEN C$
 580 IF L+LEN Y$>LEN C$ THEN GOT
O 630
 590 IF C$(L TO L+LEN Y$-1)<>Y$
THEN GOTO 630
 600 LET C$=C$( TO L-1)+X$+C$(L+
LEN Y$ TO )
 610 LET L=L+LEN X$
 620 GOTO 670
 630 IF L+LEN X$>LEN C$ THEN GOT
O 670
 640 IF C$(L TO L+LEN X$-1)<>X$
THEN GOTO 670
 650 IF X$=" VOCE " AND L>=LEN C
$-5 THEN GOTO 830
 660 LET C$=C$( TO L-1)+Y$+C$(L+
LEN X$-1 TO )
 670 NEXT L
 680 NEXT X
 690 IF C$(2)=" " THEN LET C$=C$
(2 TO )
 700 REM ACHA FRASE DE RESPOSTA
 710 LET F$=A$(R(K))( TO VAL A$(
R(K))(50 TO ))
 720 LET R(K)=R(K)+1
 730 IF R(K)>N(K) THEN LET R(K)=
S(K)
 740 PRINT "SUA RESPOSTA ANTERIO
R:"
 750 PRINT ">";I$
 760 PRINT AT 19,0;
 770 IF F$(LEN F$)="*" THEN GOTO
 810
 780 PRINT F$
 790 LET P$=I$
 800 GOTO 280
 810 PRINT F$( TO LEN F$-1),C$
 820 GOTO 790
 830 LET C$=C$( TO L-1)+" EU "
 840 GOTO 690
 850 PRINT "TABELA 1"
 860 PRINT
 870 FOR I=1 TO 40
 880 PRINT I;TAB 4;S(I);TAB 8;N(
I)-S(I)+1
 890 NEXT I
 900 PRINT "TABELA 2"
 910 PRINT
 920 FOR I=1 TO 40
 930 PRINT I;TAB 4;K$(I, TO 19)
 940 NEXT I
 950 PRINT "TABELA 3"
 960 PRINT
 970 FOR I=1 TO 7
 980 PRINT I;TAB 4;T$(I, TO 10)
 990 NEXT I
1000 PRINT
1010 PRINT "TABELA 4"
1020 PRINT
1030 FOR I=1 TO 7
1040 PRINT I;TAB 4;U$(I, TO 10)
1050 NEXT I
1060 PRINT
1070 PRINT "TABELA 5"
1080 PRINT
1090 FOR I=1 TO 122
1100 PRINT I;TAB 4;A$(I, TO 50)
1110 NEXT I
1120 STOP
1130 FOR N=1 TO 7
1140 CLS
1150 PRINT "(TABELA 3) PALAVRA-C
HAVE: ";N;" ?"
1160 INPUT I$
1170 LET T$(N)=I$
1180 LET T$(N)=T$(N)( TO 10)+STR
$ LEN I$
1190 PRINT T$(N)
1200 NEXT N
1210 RETURN
1220 REM A=INSTR(D$,W$)
1230 IF LEN W$=0 THEN GOTO 1310
1240 LET A=1
1250 LET Y=LEN W$
1260 IF Y>LEN D$ THEN GOTO 1310
1270 IF W$=D$(A TO Y) THEN RETUR
N
1280 LET A=A+1
1290 LET Y=Y+1
1300 GOTO 1260
1310 LET A=0
1320 RETURN
1330 REM PROGRAMA PARA CARREGAR
1340 CLS
1350 PRINT "ENTRE 40 PALAVRAS-CH
AVE"
1360 FOR N=1 TO 40
1370 CLS
1380 PRINT "(TABELA 2) PALAVRA-C
HAVE ";N;" ?"
1390 INPUT I$
1400 LET K$(N)=I$
1410 LET K$(N)=K$(N)( TO 19)+STR
$ LEN I$
1420 PRINT K$(N)
1430 NEXT N
1440 GOSUB 1130
1450 FOR N=1 TO 7
1460 CLS
1470 PRINT "(TABELA 4) PALAVRA-C
HAVE ";N;" ?"
1480 INPUT I$
1490 LET U$(N)=I$
1500 LET U$(N)=U$(N)( TO 10)+STR
$ LEN I$
1510 PRINT U$(N)
1520 NEXT N
1530 STOP
1540 FOR N=1 TO 122
1550 CLS
1560 PRINT "(TABELA 5) RESPOSTA
Nª ";N;":"
1570 PRINT
1580 INPUT I$
1590 PRINT I$
1600 INPUT P$
1610 IF P$<>"" THEN GOTO 1580
1620 CLS
1630 LET A$(N)=I$
1640 LET A$(N)=A$(N)( TO 50)+STR
$ LEN I$
1650 PRINT A$(N)
1660 NEXT N
1670 STOP
1680 SAVE "CARL▣"
1690 PRINT "DIGITE BREAK"
1700 PAUSE 120
1710 GOTO 1680
1720 STOP
```

Prof. Wilson José Tucci — Coordenador de Projetos Especiais da Escola Experimental Pueri Domus, o Diretor da Domus Informática. Colaborações de:

DISCO VOADOR

PARTE II

Conforme prometemos, voltamos neste número a falar em Disco Voadores.

Linha 230: testa o teclado para ver se foi pressionado o espaço, isto é, se o conteúdo do byte *Q* é 160 (ASCII «espaço» = 32, somando 128 resulta 160). Se a tecla não for o espaço, avança-se o disco, através da sub-rotina 430, e volta-se para a linha 230.
Linhas 290 — 370: se o tiro acertou o disco, então essa rotina faz o disco cair, através da sub-rotina 430, e atualiza o marcador de discos acertados.
Linhas 380 — 420: se o tiro ainda não acertou o disco, então essa rotina move o tiro, faz algumas verificações, move o disco e volta para 230, onde é lido o teclado e o ciclo se repete.
Linhas 430 — 590: essa é a sub-rotina que movimenta nosso disco. Repare na técnica que ela usa: como o disco é composto de dois traços horizontais não é preciso apagar e desenhar todo ele novamente para dar impressão de movimento, basta desenhar o disco na sua nova posição, feito pela linha 450, e depois apagar o que ficou para trás, feito pelas linhas 460 e 470.
Linhas 600 — 820: essas linhas formam a sub-rotina responsável pela explosão do disco. Dentro dessa sub-rotina vale a pena destacar:
Linhas 660 — 690: que formam a coluna de cogumelo atômico formado com a explosão do disco voador (obviamente discos voadores são movidos a energia nuclear).
Linhas 700 — 730: que desenham o chapéu do cogumelo atômico.
Linhas 740 — 780: que desenham mais alguns pontos extras na tela (marcianos ou o que você quer que sejam que foram arremessados pela explosão).
Linha 800: limpa a tela para o próximo disco.
Linhas 830 — 860: essa sub-rotina desenha a fumaça do tiro e faz o barulho do tiro, e ela também faz o barulho do disco caindo. (perceba que, na verdade, essa sub-rotina controla a sub-rotina 870, do barulho).
Linha 870: essa é uma das sub-rotinas de barulho nesse programa, ela faz cinco referências à posição de memória ***BU (— 16336),*** produzindo um barulhinho.
Linhas 890 — 930: essa é a sub-rotina que atualiza o placar, imprimindo o número de discos que já passaram, quantos tiros você já deu, quantos discos você já derrubou, e sua performance relativa, isto é, quantos discos você derrubou, dividido por quantos discos já passaram.
Linha 940: escolhe a posição vertical do disco, dada por Y1 perceba que essa posição é aleatória, mas pode ser controlada para dar um número dentro de um certo campo de valores, no caso do programa, como ele está, a posição do disco sempre vai ser entre 10 e 29, pois **RND (1)** gera sempre um número > = 0 e < 1. Multiplicando por 20 e tirando o inteiro, obtemos um número entre 0 e 19, somando 10, nossa gama de valores passa a ser entre 10 e 29, inclusive.
Linhas 950 — 970: formam uma outra sub-rotina de barulho. Esta, utilizando a rotina em Linguagem de Máquina carregada pelas linhas 980 e 990. Antes de chamar a rotina de Linguagem de Máquina, o programa em BASIC carrega alguns parâmetros para essa rotina nos bytes 6 e 7. Se você analisar o programa em Linguagem de Máquina, vai perceber que ele carrega o valor do byte 6 no registrador X, e utiliza esse valor, e o do byte 7 como contadores. Esses dois bytes geralmente não são usados pelo BASIC e, portanto, podem ser usados pelo nosso programa.

Isso encerra nosso artigo, esperamos que, além de se divertir com os discos voadores, você tenha aprendido mais algumas técnicas. ○

```
0  REM @POR DENTRO DO APPLE

         DISCO VOADOR
230 K =  PEEK (Q): IF K <  > 160 THEN
     GOSUB 430: GOTO 230
240  IF SL <  > 35 THEN 270
250 HD = HD + 1: GOSUB 840
260  GOSUB 900
270  COLOR= 15
280  IF  SCRN( 24,SL - 1) = 0 THEN
    380
290  REM ===ACERTOU!===
300  PLOT 24,SL - 1: COLOR= 0: PLOT
    24,SL
310  COLOR= 9
320  PLOT 22,SL - 3: PLOT 23,SL -
    2: PLOT 23,SL: PLOT 24,SL -
    1: PLOT 25,SL: PLOT 27,SL -
    2: VLIN SL - 3,SL - 2 AT 25
330  FOR I = 1 TO 9: GOSUB 840: NEXT
    :AC = 1
340  COLOR= 0: FOR I = 22 TO 27: VLIN
    SL - 3,SL AT I: NEXT
350 HI = HI + 1: GOSUB 900
360 YY = 1:XF = 0
370  POKE Q + 16,0: GOSUB 430
380  PLOT 24,SL - 1: COLOR= 0: PLOT
    24,SL
390 SL = SL - 1: IF SL > YF - 3 THEN
    410
400  COLOR= 0: VLIN 0,35 AT 24:SL
     = 35: POKE Q + 16,0
410  GOSUB 430
420  GOTO 230
430  REM ===DISCO===
440  COLOR= CD
450  HLIN HR + 2,HR + 3 AT YF: HLIN
    HR + 1,HR + 4 AT YF + 1
460  COLOR= 0
470  PLOT HR + 1,YF: PLOT HR,YF +
    1
480  IF AC THEN  PLOT HR + 4,YF: HLIN
    HR,HR + 5 AT YF - 1: HLIN HR
```

José Eduardo Moreira e Daniel R. Falconer

O PROGRAMA

```
 ,HR + 5 AT YF - 2
490  IF YY = 0 THEN 530
500 N = 41 - YF: GOSUB 950
510 XF = XF +  INT ( RND (1) * 3)
     - 1: IF HR + XF < 19 THEN X
    F = 1
520  IF HR + XF > 25 THEN XF =  -
    1
530 HR = HR + XF:YF = YF + YY: IF
    HR > 35 THEN 560
540  IF YF > 38 THEN  GOSUB 600
550  RETURN
560  HLIN 36,39 AT YF + 1: HLIN 3
    7,38 AT YF
570  GOSUB 940:CD =  INT ( RND (1
    ) * 14) + 1: IF CD = CC THEN
    570
580 AD = AD + 1: GOSUB 900
590 HR = 0: RETURN
600  REM ===EXPLOSAO===
610  VTAB 21: HTAB HR: PRINT PA$:
    AC = 0
620  COLOR= 15
630  PLOT HR,37: PLOT HR + 1,38: VLIN
    35,37 AT HR + 2: PLOT HR + 2
    ,39: VLIN 38,39 AT HR + 3: VLIN
    36,37 AT HR + 4: PLOT HR + 5
    ,37
640  POP : PRINT G$;G$;G$;G$
650  COLOR= CD
660  FOR I = 39 TO 6 STEP  - 6
670  HLIN 19,29 AT I - 1
680  HLIN 19,29 AT I: GOSUB 870
690  NEXT
700  HLIN 29,33 AT 2: HLIN 20,23 AT
    3: HLIN 28,34 AT 3: HLIN 18,
    24 AT 4: HLIN 27,35 AT 4: HLIN
    17,35 AT 5: HLIN 16,35 AT 6:
     HLIN 15,35 AT 7
710  HLIN 16,36 AT 8: HLIN 13,37 AT
    9: HLIN 12,38 AT 10: HLIN 11
    ,38 AT 11: HLIN 11,38 AT 12:
     HLIN 10,38 AT 13: HLIN 10,3
    8 AT 14: HLIN 10,37 AT 15
720  HLIN 10,37 AT 16: HLIN 10,37
     AT 17: HLIN 11,14 AT 18: HLIN
    16,36 AT 18: HLIN 12,13 AT 1
    9: HLIN 16,34 AT 19: HLIN 16
    ,34 AT 20: HLIN 17,34 AT 21
730  HLIN 31,33 AT 22
740  FOR I = 1 TO 22
750  IF I -  INT (I / 2) * 2 = 0 THEN
     COLOR= CD
760  IF I -  INT (I / 2) * 2 = 1 THEN
     COLOR= CC
770  FOR J = 1 TO 2 * I: NEXT
780 HR =  INT ( RND (1) * 40):YF =
     INT ( RND (1) * 15) + 20: GOSUB
    870: GOSUB 870: PLOT HR,YF: NEXT

790  COLOR= 0
800  FOR I = 0 TO 39: HLIN 0,39 AT
    I: FOR J = 1 TO 9: NEXT : NEXT
    : FOR J = 1 TO 222: NEXT
810  FOR I = 1 TO 500: NEXT
820  POP : GOTO 160
830  REM ===BARULHO===
840  COLOR= 9: HLIN 23,25 AT 35: HLIN
    22,23 AT 34: HLIN 25,26 AT 3
    4
850  FOR J = 1 TO 10: GOSUB 870: NEXT

860  COLOR= 0: HLIN 23,25 AT 35: HLIN
    22,26 AT 34: RETURN
870 S =  PEEK (BU) -  PEEK (BU) -
     PEEK (BU) -  PEEK (BU) -  PEEK
    (BU)
880  RETURN
890  REM ===PONTOS===
900  VTAB 24: HTAB 4: PRINT AD;: HTAB
    14: PRINT HD;
910  HTAB 24: PRINT HI;: IF  NOT
    HI THEN  RETURN
920  HTAB 34: PRINT  INT ((100 *
    HI) / AD);"%";
930  RETURN
```

Criptografia & Decriptografia

T-Kriptógrafo e T-Dekriptógrafo
Memória Ocupada = 2311
Soma Sintática = 42833
Nível =
Computadores = TK-85 e compatíveis

Renato da Silva Oliveira

Como a própria palavra já indica, a ***Criptografia*** (KRIPTOS = OCULTO; GRAPHEIS = ESCRITA), consiste fundamentalmente numa escrita obscura no sentido de que poucas pessoas a entendem (não estou falando do economês . . .). Normalmente a criptografia é empregada com o objetivo de tornar ainda mais sigilosos, textos de caráter secreto. Por exemplo, as fichas do SNI são muito bem criptografadas, de modo a tornar praticamente impossível, para qualquer pessoa que as veja, decifrá-las sem saber de antemão ***como***

Ao contrário do que acontecia no passado, atualmente a criptografia não se restringe a aplicações por serviços de inteligência ou por organizações militares. Apesar de continuar a ser fundamentalmente aplicada nessas duas áreas, hoje temos um número cada vez maior de empresas privadas utilizando-a. Os casos em que uma empresa tem o seu arquivo de "clientes potenciais", ***roubado*** de seu computador por uma concorrente são bastante comuns e, se suas fichas forem criptografadas, o ***roubo*** provavelmente terá sido inútil. Claro que sempre existe a possibilidade de que uma vez nas mãos de quem não devia estar, o criptograma (texto criptografado) possa ser descriptografado (decifrado). O trabalho de pesquisa ou estudo sobre criptogramas com intuito de decifrá-los chama-se ***CRIPTOANAGNOSE.***

Os métodos criptográficos são muito variados e a cada dia desenvolvem-se novas técnicas, cada vez mais sofisticadas. Pode-se contudo considerar como principais e básicos dois métodos, bastante utilizados na II Guerra Mundial: SUBSTITUIÇÃO e TRANSPOSIÇÃO. Adiante comentaremos, brevemente, esses dois métodos "clássicos".

SUBSTITUIÇÃO

Consiste em trocar os símbolos da escrita corrente e usual por outros previamente convencionados. Podemos considerar três tipos de substituição:

1) ***Estenografia*** — Quando os símbolos usuais são substituídos por símbolos gráficos convencionais. Exemplo: Taquigrafia, Código Morse, Código Secreto Marciano do Escoteiro Mirim, etc.

2) ***Numérica*** — Quando os símbolos usuais são substituídos por números.

3) ***Literal*** — Quando os símbolos usuais são substituídos por letras.

A substituição numérica e a literal são realizadas de acordo com uma equivalência entre os símbolos usuais e os símbolos criptográficos. De acordo com a convenção de equivalência, podemos ainda dividi-las em dois grupos:

I — ***De base (ou chave) simples:*** Quando a convenção de equivalência é única e fixa.

II — ***De base (ou chave) variável:*** Quando a convenção de equivalência possui duas partes, sendo uma fixa e outra variável.

TRANSPOSIÇÃO

Os métodos de transposição possuem descrição mais complexa e, portanto, daremos apenas um exemplo deles para não nos alongarmos demasiadamente. Vamos supor que queremos enviar a alguém a seguinte mensagem:

O homem é uma corda, atada entre o animal e o Além-do-Homem — uma corda sobre um abismo . F. NIETZSCHE

O número total de caracteres (letras, espaços e pontuações) usados é 1Ø1. Procuremos então, um número próximo a 1Ø1 e que possua um grande número de divisores. Por exemplo, o número 1Ø8. Para obtermos seus divisores, podemos nos utilizar do processo usual:

	1				
108	2				
54	2	4			
27	3	6	12		
9	3	9	18	36	
3	3	27	54	108	
1					

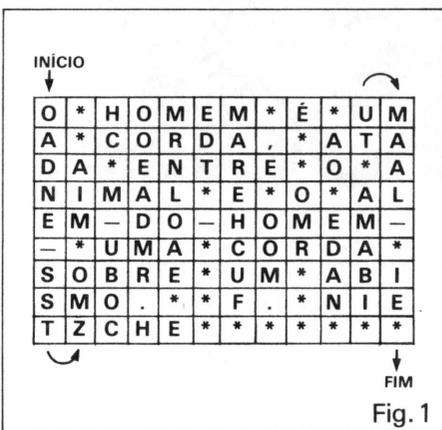
Portanto temos 1, 2, 3, 4, 6, 9, 12, 18, 27, 36, 54 e 108.

Escolheremos agora, uma matriz de linhas e colunas cujo produto do número de linhas pelo número de colunas seja igual a 108 e escreveremos nela a mensagem. Seja uma matriz de 9 linhas e 12 colunas a escolhida (Figura 1).

INÍCIO ↓

O	*	H	O	M	E	M	*	É	*	U	M
A	*	C	O	R	D	A	,	*	A	T	A
D	A	*	E	N	T	R	E	*	O	*	A
N	I	M	A	L	*	E	*	O	*	A	L
E	M	—	D	O	—	H	O	M	E	M	—
—	*	U	M	A	*	C	O	R	D	A	*
S	O	B	R	E	*	U	M	*	A	B	I
S	M	O	.	*	*	F	.	*	N	I	E
T	Z	C	H	E	*	*	*	*	*	*	*

↓ FIM

Fig. 1

Reescrevemos a mensagem lendo a matriz por colunas — no sentido indicado pelas setas na figura 1 — e não por linhas. A nova mensagem assim obtida constitui o ***CRIPTOGRAMA:***

O	A	D	N	E	—	S	S	T	*	*	A	I	M
*	O	M	Z	H	C	*	M	—	U	B	O	C	O
O	E	A	D	M	R	.	H	M	R	N	L	O	A
E	*	E	E	D	T	*	—	*	*	*	*	M	A
R	E	H	C	U	F	*	*	,	E	*	O	O	M
.	*	É	*	*	O	M	R	*	*	*	*	A	O
*	E	D	A	N	*	U	T	*	A	M	A	B	I
*	M	A	A	L	—	*	I	E	*	*	*	*	*
*	*												

A pessoa que receber o criptograma poderá decriptografá-lo facilmente se souber as características da matriz usada e a forma na qual escrevemos e "lemos" a mensagem nela. Na verdade, para obter a mensagem criptografada poderíamos ter escrito ou lido a matriz em qualquer outra direção e sentido. Por exemplo, na diagonal: (Figura 2)

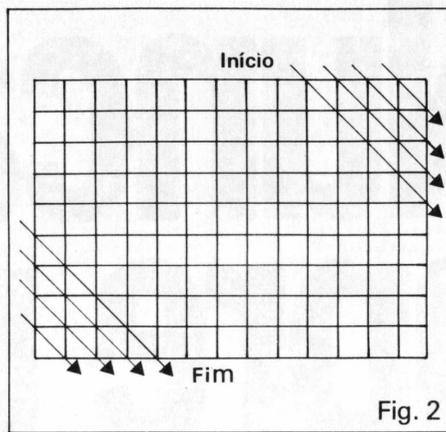



Fig. 2

Além disso, poderíamos ter escolhido qualquer outra matriz com produto de linhas por colunas igual a 1Ø8.

Como pudemos observar, esses métodos utilizam os mesmos símbolos no criptograma e na mensagem original. Apenas a posição que eles ocupam é alterada.

Para facilitar a criptografia e a decriptografia de informações, utilizam-se máquinas criptógrafas, que automaticamente alteram os símbolos ou as posições por eles ocupadas numa mensagem. Uma máquina desse tipo pode ser improvisada facilmente, trocando-se em uma máquina de escrever elétrica, a esfera (ou margarida) usual pela que contém símbolos matemáticos. Nessa situação, qualquer tecla digitada produzirá a impressão de um símbolo matemático.

Na atualidade, os métodos mais eficazes de criptografia são combinações dos métodos expostos anteriormente e, as melhores máquinas de criptografar, são os computadores.

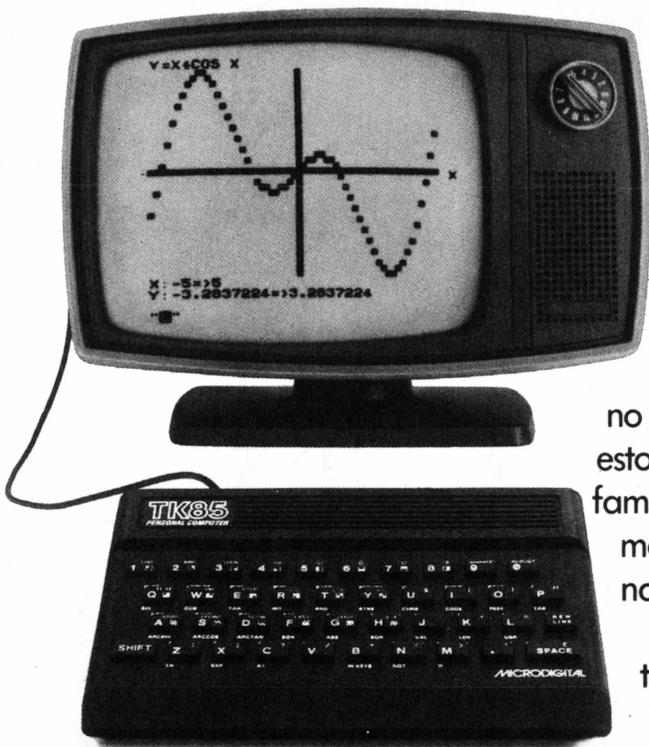
Adiante, encontra-se listado um programa que faz o TK funcionar como um Criptógrafo e Decriptógrafo (figura 3).

O processo utilizado no programa é misto (Substituição Estenográfica, numérica e literal com base variável) e, antes da introdução da mensagem, a memória ocupada pelo programa é menor que 2kBytes.

Como ilustração, vamos criptografar nele o seguinte texto:

"O que é grande no homem, é que ele é uma ponte e não um fim: o que pode ser amado no homem, é que ele é um passar e um sucumbir." F. NIETZCHE

Ao rodarmos o programa, surgirá na tela um aviso para introduzirmos a ***chave.*** Em nosso caso, essa é a base variável do criptógrafo e pode ser qualquer número, positivo ou negativo (A base fixa é a STRING Z$, construída nas linhas de 3 a 8). Introduza como →

Nunca compre uma coisa que você não vai usar.

Leve logo um microcomputador TK 85, porque ele é realmente fácil de usar: já vem com manual de instruções, que ensina, em português claro, a linguagem Basic.

A partir daí, você pode preparar seus próprios programas ou utilizar as centenas de programas que já existem no mercado, para cadastrar clientes, controlar estoques, manter em ordem o orçamento familiar, fiscalizar a conta bancária, estudar matemática, estatística, jogar xadrez, guerra nas estrelas, e o que mais você puder imaginar.

E além disso tudo, o TK 85 tem também o preço mais acessível do mercado. Peça uma demonstração.

TK 85, o micro que você pode usar.

Filiada a ABICOMP

FOX

exemplo o número 1984. Agora digite a mensagem e, quando terminá-la, NEW LINE. A seguir, o micro pedirá para introduzir *C* ou *D* para criptografar ou decriptografar a mensagem. Introduza, então o caractere *C*. No vídeo, após alguns instantes, surgirá a mensagem criptografada (figura 4) e uma pergunta: "Quer reverter a mensagem? (S/N)". Se você quiser revertê-la (decriptografá-la) introduza *S* e, nesse caso a mensagem original retornará à tela (figura 5). Caso contrário, introduzindo *N*, o TK ficará a espera de uma nova mensagem.

```
  1 REM "█-█RIPTOGRAFO  £
         █-DE█RIPTOGRAFO"
  2 FAST
  3 LET Z$=""
  4 FOR Z=0 TO 192
  5 IF Z=11 THEN LET Z=12
  6 IF Z=64 THEN LET Z=128
  7 LET Z$=Z$+CHR$ Z
  8 NEXT Z
  9 PRINT "INTRODUZA A ""CHAVE"
":"
 10 INPUT R
 11 PRINT R
 12 PAUSE 180
 13 CLS
 14 IF R<1 THEN LET R=R+65535
 15 IF R<1 THEN GOTO 14
 16 IF R>65535 THEN LET R=R-655
35
 17 IF R>65535 THEN GOTO 16
 18 RAND R
 19 LET L=INT (.5+SQR (65535*RN
D))
 20 IF L>127 THEN LET L=L-127
 21 IF L>127 THEN GOTO 20
 22 PRINT "INTRODUZA A MENSAGEM
:"
 23 INPUT A$
 24 PRINT A$
 25 PAUSE 240
 26 CLS
 27 PRINT "ENTRE C P/ CRIPTOGRA
FAR             E D P/ DECRIPTOGRAF
AR:"
 28 INPUT I$
 29 IF I$<>"C" AND I$<>"D" THEN
 GOTO 28
 30 LET O$=""
 31 IF I$="D" THEN LET O$="DE"
 32 PRINT "A MENSAGEM SERA ";O$
;"CRIPTOGRAFADA."
 33 PAUSE 180
 34 CLS
 35 LET B$=""
 36 FOR F=1 TO LEN A$
 37 LET C=CODE A$(F)
 38 IF C=11 OR (63<C AND C<128)
 OR C>192 THEN GOTO 59
 39 IF 12<=C AND C<=63 THEN LET
 C=C-1
 40 IF 128<=C THEN LET C=C-65
 41 IF I$="D" THEN LET C=C+2
 42 LET K=C+L*(1 AND I$="C")-L*
(1 AND I$="D")
 43 IF K>128 THEN LET K=K-128
 44 IF K<1 THEN LET K=K+128
 45 LET B$=B$+Z$(K)
 46 NEXT F
 47 PRINT "ESTA E A MENSAGEM TR
ADUZIDA:"
 48 PRINT B$
 49 PRINT
 50 PRINT "QUER REVERTER A MENS
AGEM?(S/N):"
 51 INPUT L$
 52 IF L$="S" THEN GOTO 56
 53 IF L$<>"S" AND L$<>"N" THEN
 GOTO 51
 54 PRINT "AGUARDO OUTRA MENSAG
EM:"
 55 GOTO 9
 56 LET A$=B$
 57 LET I$="D"
 58 GOTO 35
 59 PRINT "O CARACTERE";F;"(";A
$(F);") NAO E PERMITIDO; MODIFIQ
UE-O:"
 60 INPUT N$
 61 LET A$(F)=N$
 62 GOTO 32
```

Fig. 3

```
ESTA E A MENSAGEM TRADUZIDA:
T█VZJ█J█LWFSIJ█ST█MTRJR3█J█VZJ
█JQJ█J█ZRF█UTSYJ█J█SFT█ZR█KNR<█
T█VZJ█UTIJ█XJW█FRFIT█ST█MTRJR3█J
█VZJ█JQJ█J█ZR█UFXXFW█J█ZR█XZHZ
RGNW4████████K4█SNJY█HMJ

QUER REVERTER A MENSAGEM?(S/N):
```

Fig. 4

```
ESTA E A MENSAGEM TRADUZIDA:
T█VZJ█J█LWFSIJ█ST█MTRJR3█J█VZJ
█JQJ█J█ZRF█UTSYJ█J█SFT█ZR█KNR<█
T█VZJ█UTIJ█XJW█FRFIT█ST█MTRJR3█J
█VZJ█JQJ█J█ZR█UFXXFW█J█ZR█XZHZ
RGNW4████████K4█SNJY█HMJ

QUER REVERTER A MENSAGEM?(S/N):
ESTA E A MENSAGEM TRADUZIDA:
O QUE E█ GRANDE NO HOMEM, E█ QUE
 ELE E█ UMA PONTE E NAO UM FIM:
O QUE PODE SER AMADO NO HOMEM, E
█ QUE ELE E█ UM PASSAR E UM SUCU
MBIR.          F. NIETZCHE

QUER REVERTER A MENSAGEM?(S/N):
```

Fig. 5

Criptoanagnose

Renato da Silva Oliveira

Caros leitores, mais uma vez nós somos forçados a recorrer a inteligência, perspicácia, determinação e boa vontade de todos vocês. Temos novamente um problema de ***difícil*** solução em nossas mãos.

Tudo começou quando, no mês de dezembro do ano passado, uma das secretárias resolveu tirar férias. Como havia muito trabalho em andamento haveria um grande desfalque em nosso "time" o Pierluigi teve a ***mui infeliz e insana*** idéia de trazer para MICROMEGA uma secretária "free-lancer".

Assim, ocorreu, e no dia 13 de Dezembro tivemos o prazer, infinitesimal momentâneo e visual, de conhecer a nova "secretária". Nos primeiros instantes ela nos pareceu muito boa, sob todos os aspectos. Ela é alta, loira, de olhos azuis, corpo escultural, fala macia e pele queimada do sol. Realmente, poderia ser facilmente "Miss Universo", tal a sua exuberância.

Tudo teria sido uma maravilha se a natureza tivesse dado a seu cérebro ao menos um milésimo do aprimoramento que deu a seu corpo! Acreditem! Ela é a comprovação definitiva das Leis de ***Murphy!*** Se vocês estão pensando que seu nome começa com ***D*** e termina com ***A*** acertaram. Ela era a própria ***DINORÁ!***

No primeiro, e felizmente único dia em que "trabalhou" conosco, apenas no horário do almoço, ela conseguiu proezas incríveis! Vocês já ouviram falar da ***PERVERSIDADE UNIVERSAL*** DA MATÉRIA? Pois é. Ela encontra-se concentrada na Dinorá. Como afirma o segundo corolário da primeira lei de Murphy: "Se há possibilidade de diversas coisas saírem erradas, então sairá errada aquela que causar maior prejuízo". Isso nunca falha quando a Dinorá está por perto.

Vocês podem imaginar alguém que pára diante de uma caixa de fitas e começa a passá-las uma por uma pelo "Túnel da Virgindade" (vulgo TWINGO)? Pois foi assim que a Dinorá conseguiu apagar 11 fitas nas quais estavam gravados 11 longos programas de mais de 8 kBytes cada!

Não bastasse isso, quando por acaso ela conseguiu fazer funcionar um TK-82, enganchou seus longos cabelos dourados numa impressora que estava acoplada a ele. Essa foi a última vez que o TK e a impressora foram vistos inteiros!

Como se diz ***"ABISUM ABISUS"*** (o abismo chama o abismo), e quando ela livrou-se do que restou da impressora do TK, acabou tropeçando no cabo que liga a impressora do nosso TRS-80 ao próprio e ambos acabaram no chão!

Ela ainda arrumou tempo para derrubar uma garrafa térmica em cima de uma prancheta onde estavam os originais da revista número 7 (que tiveram que ser totalmente refeitos); levar ao chão a nossa árvore de Natal que havíamos montado pela manhã; misturar os arquivos de "Cartas dos Leitores" e, por uma fatalidade, receber uma carta e um telegrama do Sr. Nabor Rosenthal.

Como vocês sabem, o Sr. Nabor, além de ser assinante de Microhobby, é grande amigo de todos nós aqui da redação e freqüentemente nós recebemos dele correspondências com contribuições à revista, com mensagens pessoais. Devido ao fato de as cartas do Sr. Nabor conterem algumas vezes informações sigilosas e de grande importância para a sobrevivência de Microhobby, ele a algum tempo as tem criptografado antes de enviá-las a nós.

Esses textos são criptografados com o programa "T-KRIPTÓGRAFO & T-DE KRIPTÓGRAFO" e a chave para decriptografá-los nos é remetida por telegrama.

Ocorre que o telegrama, a Dinorá deixou cair num vaso sanitário e a carta ela deixou, felizmente cair pela janela do banheiro. Graças a ação de uma boa lufada de vento, a carta caiu no capô de meu carro quando eu acabara de estacionar!

Agora, três grandes incógnitas perturbam minha mente:

1) Qual o número ***chave*** utilizado pelo Sr. Nabor para criptografar a carta?

2) O que está escrito na mensagem?

3) Como a Dinorá conseguiu o emprego de secretária?

O Sr. Nabor encontra-se agora em viagem por algum lugar do Mundo e nós não conseguimos localizá-lo. Dessa forma, resolvemos pedir auxílio à esta seção. Claro que não estamos esperando resposta à terceira questão, pois, já nos convencemos que isso seria impossível! Nem mesmo a resposta que lhes passou pela mente justifica a pergunta!

Para facilitar o trabalho temos a seguinte informação a transmitir: O Sr. Nabor sempre utiliza um número de quatro algarismos como chave.

Agradecemos imensamente o que vocês puderem fazer para amenizar nossa curiosidade e ficamos aguardando um programa para o TK, que solucione as primeiras duas questões.

Adiante encontra-se a mensagem criptografada. ○

CRIAÇÃO DE COELHOS

"Antes de apresentar a solução para o problema questionado, é preciso esclarecer alguns detalhes:

A série numérica sugerida é conhecida como série de Fibonacci, cujo termo desejado é igual a soma dos dois termos anteriores.

Exemplo: termos iniciais — 0 e 1
0,1,1,2,3,5,8,13. . .

Quanto aos valores apresentados, o número de coelhos da 15ª ninhada, segundo a série, é de 4181 e não 418 como foi apresentado.

As respostas para o teste proposto são:

A) o valor do próximo termo da série é 6765

B) são 46.368 coelhos, o valor depois da 20ª ninhada

C) pode-se começar a série com quaisquer outros dois valores iniciais, e o programa teria que ser modificado conforme o programa em anexo (figura 2) de nome "NOVSER".

d) o programa principal segue em anexo (figura 1), com o nome de "SÉRIE".

José Rosa da Fonseca

NOTA DA REDAÇÃO I

Terceira Lei de Murphy

"Em qualquer conjunto de dados, os erros estão contidos naqueles números que são obviamente corretos e isentos de qualquer necessidade de conferência".

Corolário 1

"Ninguém a quem você pedir auxílio verá o erro".

Corolário 2

"Qualquer intrometido inoportuno que aparecer e bater com os olhos nos dados, vai encontrar o erro imediatamente".

NOTA DA REDAÇÃO II

O Quebra-Cabeça da Microhobby nº 8 continha, na verdade, dois quebra-cabeças. O primeiro era o que nós propusemos conscientemente. O segundo involuntariamente, a partir de erros de impressão que escaparam por entre nossas muitas revisões.

O trecho incorreto era o do segundo parágrafo da segunda coluna, na página 34, onde estava escrito:

"Por exemplo: se você começar com um macho e uma fêmea, na décima ninhada, terá 377 coelhos e na décima quinta, *terá 418 coelhos".*

O correto teria sido:

*"Por exemplo: se você começar com um macho e uma fêmea, na dé*cima primeira *ninhada, terá 377 coelhos e na* décima sexta, *terá 4181 coelhos".*

Apesar do erro, José Rosa da Fonseca *conseguiu solucionar os quebra-cabeças (o que propusemos conscientemente e o involuntário). Aqui está sua solução que devido as circunstâncias desfavoráveis, não se encontra corretamente justificada. Não é necessário supor que os coelhos morrem!). Recebemos também, uma solução correta do* Antonio Cesar Lettieri *(figura 3).*

```
   1 REM "SERIE"
   2 REM JOSE R. FONSECA/84
   5 CLS
  10 LET A=2
  15 LET B=3
  20 GOSUB 100
  25 PRINT AT 18,0;"DIGITE O NUM
ERO DA NINHADA CUJO";AT 19,5;"VA
LOR E DESEJADO"
  30 INPUT N
  35 CLS
  40 FAST
  45 FOR I=1 TO N
  50 LET D=A+B
  55 LET A=B
  60 LET B=D
  65 NEXT I
  70 SLOW
  75 GOSUB 100
  80 PRINT AT 8,0;"NA";AT 8,4;N;
AT 8,7;;"A. NINHADA HAVERA";AT 1
0,7;D;AT 10,13;"COELHOS"
  85 STOP
 100 PRINT AT 2,0;"CRIACAO DE CO
ELHOS"
 105 RETURN


OBS:   PARA  OBTER  A  RESPOSTA
       PRETENDIDA ORIGINALMENTE
       SUBSTITUA AS LINHAS 10 E
       15 POR:

       10 LET A=1
       15 LET B=2
```

Figura 1

```
   1 REM "NOVSER"
   2 REM JOSE R. FONSECA/84
   3 CLS
   4 GOSUB 100
   5 PRINT AT 5,0;"DIGITE OS DOI
S PRIMEIROS NUMEROS";AT 7,3;"DA
SERIE DESEJADA"
   6 INPUT P
   7 LET A=P
   8 INPUT S
   9 LET B=S
  10 PRINT AT 9,5;"CONFIRME OS D
OIS PRIMEIROS";AT 13,0;"NUMEROS"
;AT 13,10;"(S) OU (N)"
  11 PRINT AT 16,5;A;AT 16,10;B
  12 INPUT N$
  13 IF N$="N" THEN GOTO 3
  14 IF N$<>"S" THEN GOTO 12
  15 CLS
  25 PRINT AT 18,0;"DIGITE O NUM
ERO DA NINHADA CUJO";AT 19,5;"VA
LOR E DESEJADO"
  30 INPUT N
  35 CLS
  40 FAST
  45 FOR I=1 TO N
  50 LET D=A+B
  55 LET A=B
  60 LET B=D
  65 NEXT I
  70 SLOW
  75 GOSUB 100
  80 PRINT AT 8,0;"NA";AT 8,4;N;
AT 8,7;;"A. NINHADA HAVERA";AT 1
0,7;D;AT 10,13;"COELHOS"
  85 STOP
 100 PRINT AT 2,0;"CRIACAO DE CO
ELHOS"
 105 RETURN
```

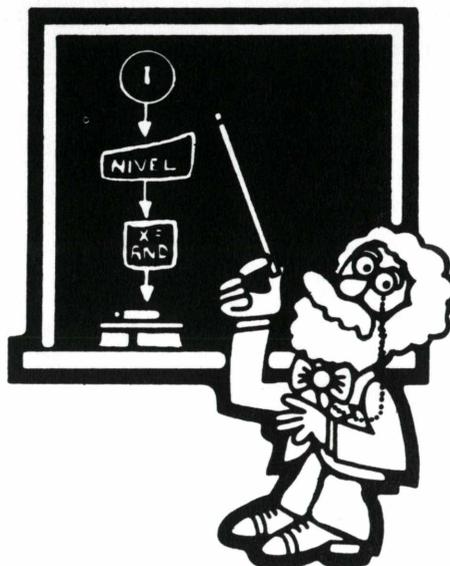
Figura 2

```
  10 REM CRIACAO DE COELHOS
  20 SCROLL
  30 PRINT "QUANTAS NINHADAS?"
  40 SCROLL
  50 INPUT PARA
  60 PRINT "OK, ";PARA;" NINHADA
S"
  70 SCROLL
  80 LET A=2
  90 LET B=C
 100 FOR I=1 TO PARA
 110 LET C=A+B
 120 PRINT "NIHADA Nº ";I;": ";C
" COELHOS"
 130 SCROLL
 140 LET A=B
 150 LET B=C
 160 NEXT I


OBS:   PARA  OBTER  A  RESPOSTA
       PRETENDIDA ORIGINALMENTE
       SUBSTITUA AS LINHAS 80 E
       90 POR:

       80 LET A=1
       90 LET B=2
```

Figura 3

CURSO DE   TK

aula 12

Pierluigi Piazzi e Flávio Rossini

O gravador K-7

Você deve ter notado como não é muito prático ter que digitar um programa "grande", especialmente quando isto já foi feito diversas vezes . . . Felizmente o TK possibilita a gravação de programas em fita K-7 comum para posterior recuperação. Para colocar um programa no gravador, utiliza-se a Key-word SAVE (letra S), dando-se um nome ao programa (o qual deve ser colocado entre aspas). Suponha que tenhamos um programa chamado CHOCOLATE. Conecte então a saída MIC do seu TK à entrada MIC do seu gravador, ajuste o volume para aproximadamente 70 por cento do nível máximo e, se tiver controle de tonalidade, coloque-o no mais agudo possível. A seguir pressione as teclas do gravador para gravar e digite o seguinte comando direto:

```
SAVE "CHOCOLATE"
```

e NEW LINE. A tela ficará "esquisita" por alguns instantes; quando ela voltar ao "normal", a gravação está terminada. É recomendável que, por segurança, você grave um mesmo programa diversas vezes na fita.

Para recuperar o programa da fita K-7 a operação também é simples.

Suponha que você tenha realmente gravado um programa chamado CHOCOLATE. Digite NEW para limpar a memória, volte a fita e conecte a entrada EAR do seu TK à saída EAR do seu gravador e, com os controles de volume e tonalidade idênticos aos da operação de SAVE, coloque o seu gravador para "tocar", após digitar o seguinte comando direto:

```
LOAD "CHOCOLATE"
```

e NEW LINE (o LOAD está na tecla J). Se após a tela "esquisita" aparecer a mensagem Ø / Ø, você conseguiu recuperar o programa; caso contrário, repita a operação.

Aqui vai uma dica: experimente digitar o seguinte:

```
10 PRINT "LOVE IS THE ANSWER"
20 GOTO 10
30 SAVE "LOV█"
40 RUN
```

Prepare o gravador para um SAVE e digite o seguinte comando direto:

```
GOTO 30
```

O programa será armazenado na fita . . . mas, ao carregá-lo de volta, ele começa a sua execução sem ser necessário digitar RUN e NEW LINE; experimente então voltar a fita e fazer:

```
LOAD "LOVE"
```

PEEK, POKE e USR: Interfaces do BASIC com a Linguagem de Máquina

Nesta altura do curso, você pode olhar para o teclado do TK e verificar que tudo já foi explicado, com exceção de uma Key-word e duas funções: POKE (tecla 0), PEEK (tecla 0) e USR (tecla L). Para poder dar uma *noção* do que elas significam, é necessário primeiro ter uma idéia do que é Linguagem de Máquina, conceito de microprocessador e bites de memória. Para tanto, sugerimos a leitura da aula 2 do curso de Assembly publicado na revista número 3. Baseados nesses

conceitos, vamos então explicar resumidamente estes últimos comandos do BASIC.

A Key-word POKE permite colocar um número diretamente numa posição qualquer de memória. Naturalmente, este número deve estar compreendido entre 0 e 255 e a posição de memória entre 0 e 65535. Entretanto, as posições de 0 a 16383 são ocupadas por memória ROM, que é inalterável, ou por espaços vazios. Assim, um POKE nessas memórias não produz efeito algum. A título de exemplo, faremos dois pequenos truques usando o POKE; pelo "mapa" de memória do TK (veja revista número 4) vimos que os programas em BASIC começam sempre na memória 16509; sabe-se que, normalmente, não é possível em BASIC ter a linha 01. Mas com POKE podemos consegui-la. De fato, experimente digitar:

```
1 REM PIAZZI-ROSSINI
2 PRINT "ALL YOU NEED IS LOV
E"
3 RUN
```

A seguir, faça o comando direto:

```
POKE 16510,0
```

Agora, faça um LIST e veja o que ocorreu com a linha 1! Tente retirá-la do programa e você verá que não será possível, pois o BASIC não está "acostumado" a usar linhas 0 . . . Qual a vantagem disto? Ora, você pode fazer um programa, colocar a linha 0 com seu nome e ninguém poderá retirá-lo. Pode surgir agora a seguinte dúvida: se os programas começam na memória 16509, por que o POKE é feito na memória 16510? A explicação é simples: em BASIC, podemos ter linhas com número maior do que 255; sendo assim, dois bytes são reservados para armazenar o número da linha (no caso teremos as memórias 16509 e 16510, ambas com o número zero; experimente modificar a memória 16509 usando POKE . . .). No manual do TK, você poderá encontrar maiores detalhes sobre como cada linha e cada variável é armazenada na memória.

O segundo truque que iremos apresentar possibilita a utilização das linhas 22 e 23 da tela para efeitos de PRINT ou PRINT AT. Na memória 16418 está armazenado um número (inicialmente pelo programa interpretador) que diz quantas linhas são reservadas para a região de edição. Normalmente, essa memória contém o número 2. Basta então alterá-la para 0; repare:

```
1 POKE 16418,0
2 PRINT AT 22,5;" LOVE IS"
3 PRINT AT 23,5;" THE ANSWER"
```

No entanto, este truque tem uma pequena limitação: não possibilita a utilização da Key-word INPUT. Assim, se por acaso você for usá-la, "corrija" antes a memória 16418 com a instrução:

```
POKE 16418,2
```

O que acontece se colocarmos nesta memória um número maior do que 2? (veja na revista número 2, página 24.

Vamos agora dar uma olhada na função PEEK; ela fornece como saída o conteúdo de uma determinada memória (naturalmente entre 0 e 65535). Assim, se ao ligar o computador você fizer:

```
PRINT PEEK 16418
```

O computador mostrará o número 2 na tela, pois, inicialmente duas linhas são reservadas para edição. Se o seu programa tiver uma linha 1, o que você espera obter com um PEEK 16510? E com um PEEK 16509?

Note que você pode inclusive, fazer PEEK na memória ROM. De fato, para obter o conteúdo da ROM na tela basta fazer:

```
10 SLOW
20 FOR I=0 TO 16383
30 SCROLL
40 PRINT PEEK I
50 NEXT I
```

Como exemplo da função PEEK vamos apresentar um pequeno programa que renumera as linhas de um dado programa em BASIC, sem alterar entretanto os GOTOs ou GOSUBs (isto também seria possível, mas o programa é bem mais complexo). A explicação do funcionamento desse programa é razoavelmente complicada e foge dos objetivos desse curso. O programa é aqui apresentado apenas para dar uma idéia do "potencial" da função PEEK.

```
9000 STOP
9005 REM
9010 FAST
9015 PRINT "NUMERO DA LINHA INIC
IAL"
9020 INPUT NIR
9025 PRINT NIR
9030 PRINT "NOVA LINHA INICIAL ?
"
9035 INPUT NLI
9040 PRINT NLI
9045 PRINT "INTERVALO ENTRE AS L
INHAS ?"
9050 INPUT DTL
9055 PRINT DTL
9056 CLS
9057 LET IM=16508
9060 LET NX=PEEK (IM+1)*256+PEEK
 (IM+2)
9070 IF NX=9000 THEN GOTO 9110
9075 POKE (IM+1),INT (NLI/256)
9080 POKE (IM+2),NLI-256*INT (NL
I/256)
9085 LET NLI=NLI+DTL
9090 LET IM=IM+5
9095 IF PEEK IM=118 THEN GOTO 90
60
9100 LET IM=IM+1
9105 GOTO 9095
9110 LIST
```

Coloque seu programa antes da linha 9000 e forneça ao RENUMBER a linha inicial de seu programa, a nova linha inicial e a distância que você deseja entre as linhas. Este programa é muito útil quando não há mais "espaço" entre linhas para colocar novas instruções ou para tornar as listagens mais "elegantes".

Para finalizar este curso, vamos dar uma idéia sobre a função USR. Ela seria como um GOSUB que em vez de "ir" para uma linha d eprograma, vai direto para uma posição de memória executar uma sub-rotina escrita em Linguagem de Máquina. Apenas para dar uma demonstração da potencialidade da Linguagem de Máquina, experimente o seguinte programa que "enche" a tela com um determinado caráctere:

```
1 REM YUX/X
2 POKE 16516,215
3 POKE 16518,253
4 LET L=USR 16514
```

Os caracteres na linha REM significa um pequeno programa em Linguagem de Máquina; o caractere que é colocado na tela é o segundo caractere da linha REM. Experimente modificá-lo . . . Compare a velocidade deste programa (mesmo em SLOW!) com o seguinte programa em BASIC:

```
3 PRINT "W"
4 GOTO 3
```

Poderia surgir agora a seguinte pergunta: existe alguma vantagem em se programar em Linguagem de Máquina? A resposta é afirmativa: com ela, os programas são muito mais rápidos e podemos ter uma visualização e controle maior do sistema. Em contrapartida, suas instruções são mais simples e limitadas e exigem um cuidado maior na hora da programação.

Não nos aprofundaremos mais, neste curso, na Linguagem de Máquina. Estas idéias foram aqui apresentadas apenas para tentar dar uma noção do que é essa linguagem, com o intuito de tentar esclarecer os conceitos que apresentaremos a seguir. Note entretanto, que um estudo detalhado da Linguagem de Máquina é relativamente complexo e o pequeno resumo que apresentamos aqui está longe de mostrar toda a sua potencialidade. Caso você tenha se entusiasmado pela velocidade e magia da Linguagem de Máquina, sugerimos a leitura do curso de Assembly publicado nesta revista a partir do número 2 ou então, para um estudo mais profundo, o seguinte livro:

"INTRODUÇÃO À LINGUAGEM DE MÁQUINA PARA O TK — ASSEMBLY Z80"

VOL. I de Flávio Rossini, ED. MODERNA — MICROMEGA

Caso você queira se aprofundar no BASIC, sugerimos a leitura e estudo dos vários programas publicados nesta revista ou ainda os seguintes livros:

"CURSO DE JOGOS" de Fábio Rendelucci

"Coleção de programas vol. I" de Carlos E. R. Salvato.

"Coleção de programas vol. II" de Tanios Hamzo.

COMO COLABORAR COM MICROHOBBY

A Revista MICROHOBBY foi criada para servir de intercâmbio entre os leitores que participam do mágico mundo da computação.

A característica realmente inovadora do computador pessoal, está em transformar cada consumidor num criador. Aproveite sua criatividade e envie suas colaborações recebendo remuneração a título de DIREITO AUTORAL.

A maneira ideal de nos enviar o material a ser publicado obedece às seguintes normas:

1. Nunca esqueça de colocar o nome completo, telefone, endereço e número de sua assinatura em **todo** material enviada a nós, sejam listagens de impressora, fitas, envelope, carta ou qualquer outro material.

2. Envie a listagem de programa **datilografada** ou, melhor ainda, tirada na impressora do computador.

3. Coloque sempre uma linha REM com o nome do autor e o título do programa.

4. Envie uma fita com o programa gravado **algumas vezes** (se possível em gravadores diferentes).

5. Na fita, gravar **com microfone** (em viva voz), algumas instruções úteis: **nome completo** e **endereço do remetente.**

6. Quando o programa for adaptado e/ou traduzido de outra revista, citar a fonte (autor original, data de publicação, nome da revista e todos os detalhes que houver referente à publicação).

7. Anexar ao material, uma carta autorizando a publicação por parte da revista e assumindo a responsabilidade pela autoria do material e/ou adaptações. Nesta carta, para agilizar a remuneração, podem constar os dados da conta corrente onde daremos o depósito correspondente aos direitos autorais.

8. O material **não utilizado não será devolvido,** ficando a critério da redação a decisão final sobre sua publicação.

9. O material deve ser enviado para:

MICROMEGA PUBLICAÇÕES E MATERIAL DIDÁTICO
SEÇÃO PROGRAMAS DO LEITOR
Cx. Postal 54096
CEP 01296

Flavio Rossini

Instruções lógicas e Operações com bits

As instruções lógicas são instruções que mexem diretamente com os bits. Vamos inicialmente dar uma pequena explicação das principais operações que são AND, OR, XOR e NOT. Para isto, basta associar aos bits a "teoria dos conjuntos" da seguinte maneira: ao número 1 associamos o conjunto Universo e ao número Ø o conjunto vazio; a operação AND seria a operação de *intersecção* de conjuntos, a operação OR corresponde à *união* e a função NOT ao *complemento* (Figura 1).

Na Linguagem de Máquina, existem instruções que fazem estas operações *bit a bit,* sempre tendo o *acumulador* como sede dos resultados.

A função AND só resulta 1 se os 2 bits correspondentes forem 1, caso contrário resulta Ø; a função OR dá resultado Ø apenas se os dois bits correspondentes forem Ø, senão ela fornece o número 1 e a função NOT tem apenas UM operando e coloca Ø onde havia 1 e 1 onde havia Ø. Quanto à função XOR, ela fornece o resultado 1 somente se os bits correspondentes forem *diferentes.*

Podemos ter as seguintes instruções (figura 2):

AND	OR	NOT
1 AND 1 = 1	**1 OR 1 = 1**	**NOT Ø = 1**
1 AND Ø = Ø AND 1 = Ø	**1 OR Ø = Ø OR 1 = 1**	**NOT 1 = Ø**
Ø AND Ø = Ø	**Ø OR Ø = Ø**	

Fig. 1

INSTRUÇÃO	CÓDIGO	INSTRUÇÃO	CÓDIGO	INSTRUÇÃO	CÓDIGO
AND A	'A7'	OR A	'B7'	XOR A	'AF'
AND B	'AØ'	OR B	'BØ'	XOR B	'A8'
AND C	'AL'	OR C	'BL'	XOR C	'A9'
AND D	'A2'	OR D	'B2'	XOR D	'AA'
AND E	'A3'	OR E	'B3'	XOR E	'AB'
AND H	'A4'	OR H	'B4'	XOR H	'AC'
AND L	'A5'	OR L	'B5'	XOR L	'AD'

Fig. 2

AND registro: realiza a operação AND do registro com o acumulador colocando o resultado no acumulador.

OR registro: idem com relação à operação OR.

XOR registro: idem com relação à operação XOR.

Estas três instruções afetam o flag do CARRY (deixando-os em Ø (zero) e todos os outros flags vistos até aqui; assim, uma maneira de "zerar" o acumulador e simultaneamente "zerar" o flag de CARRY é usando a instrução XOR A.

O único registro, como você deve estar esperando, que permite a operação NOT é o acumulador, com a seguinte instrução:

CPL (abreviação de COMPLEMENT) (Código '2F')

ela NÃO altera o flag de CARRY e nem os demais FLAGS vistos até o presente momento.

Vamos fazer alguns exemplos para fixar melhor o funcionamento destas instruções (figuras 3, 4, e 5).

Execute estes programas modificando eventualmente os dados para fixar bem o funcionamento das operações lógicas.

Temos ainda as instruções que fazem essas operações utilizando o conteúdo da memória (endereçada por HL) ou com um dado de 8 bits diretamente (figura 6).

Como as instruções AND e OR afetam todos os FLAGS, elas podem ser utilizadas para decidir se um dado número no acumulador é positivo, negativo ou zero quando queremos efetuar algum "salte", sem necessitar utilizar a instrução CP. Por exemplo, suponha que em dado ponto do programa você deseja "saltar" para a memória 30050 se o conteúdo do registro B for positivo (figura 7).

Instruções que afetam os bits

Vamos agora aprender instruções que modificam INDIVIDUALMENTE os valores dos bits dos registros ou conteúdo da memória indicada por HL. Talvez no momento você não veja muita utilidade para estas instruções, mas, se você se aprofundar na Linguagem de Máquina verá como elas poderão ajudá-lo. Assim, temos as instruções de SET para colocar dado bit em 1 e RES para colocar dado bit em 0. Devido ao grande número de combinações possíveis vamos montar uma tabela (figura 8 A e B).

Observe na tabela da figura 8A e 8B que os bits de cada registro (que são 8) são contados de 0 a 7. Assim, por exemplo, se você quiser colocar em 1 o bit 5 do registro E:

```
SET 5,E        'CBEB'
```

para colocar em 0 o bit 6 da memória indicada por HL:

```
RES  6, (HL)  'CBB6'
```

Façamos um exemplo (figura 9): vamos chamar uma sub-rotina na memória 30000, (teremos então BC = '7530' = "0111010100110000") e vamos fazer B = '00' e C = 'FF' setando e resetando os bits necessários, tendo então como saída (ao apertar XF no HEXAMEM) o número 255 ('00FF').

Caso queiramos "testar" um dado bit de um registro ou memória endereçada por HL para saber se ele é zero ou um, podemos usar a instrução Bit que coloca o *complemento* do dado bit no flag Z do registro F. Assim, por exemplo, se você quiser saltar para um dado endereço (ou chamar uma sub-rotina) baseado na CONDIÇÃO que o bit 5 do registro E seja *zero* (0) basta fazer:

AND

```
30000   LD     A,'AF'        '3EAF'
30003   LD     B,'5A'        '065A'
30004   AND    B             'A0'     ; (ou AND A,B)
30005   LD     C,A           '4F'
30006   LD     B,0           '0600'
30008   RET                  'C9'

temos    A = "10101111"   resultando A = "0000 1010" = '0A' = 10
         B = "01011010"
```

Fig. 3

OR

```
30000   LD     A, '82'       '3E82'
30002   LD     E, '27'       '1E27'
30004   OR     E             'B3'     ; (ou OR A,E)
30005   LD     C,A           '4F'
30006   LD     B,0           '0600'
30008   RET                  'C9'

temos    A = "10000010" resultando A = "10100111" = 'A7' = 167
         E = "00100111"
```

Fig. 4

NOT

```
30000   LD     A,'F0'        '3EF0'
30002   CPL                  '2F'     ; (ou CPL A)
30003   LD     C,A           '4F'
30004   XOR A                'AF'
30005   LD     B,A           '47'
30006   RET                  'C9'
temos    A = "11110000" resultando A = "00001111" = "OF' = 15
```

Fig. 5

INSTRUÇÃO	CÓDIGO	
AND (HL)	'A6'	; (ou AND A, (HL))
OR (HL)	'B6'	; (ou OR A, (HL))
XOR (HL)	'AE'	; (ou XOR A, (HL))
AND dado	'E6' + 1byte P/ o dado	; (ou ND A, dado)
OR dado	'F6' + 1byte P/ o dado	; (ou OR A, dado)
XOR dado	'EE' + 1byte P/ o dado	; (ou XOR A, dado)

Fig. 6

```
LD          A,B
OR          A          ; não altera o acumulador, apenas os FLAGS
                         (OR A,A)
JP     P    30050
```

Fig. 7

SET

Bit \ Reg	A	B	C	D	E	H	L	(HL)
0	'CBC7'	'CBC0'	'CBC1'	'CBC2'	CBC3'	'CBC4'	'CBC5'	CBC6'
1	'CBCF'	CBC8'	'CBC9'	'CBCA'	'CBCB'	'CBCC'	CBCD'	'CBCE'
2	'CBD7'	'CBD0'	'CBD1'	'CBD2'	'CBD3'	'CBD4'	'CBD5'	'CBD6'
3	'CBDF'	'CBD8'	'CBD9'	'CBDA'	'CBDB'	'CBDC'	'CBDD'	'CBDE'
4	'CBE7'	'CBE0'	'CBE1'	'CBE2'	'CBE3'	'CBE4'	'CBE5'	'CBE6'
5	'CBEF'	'CBE8'	'CBE9'	'CBEA'	'CBEB'	'CBEC'	'CBED'	'CBEE'
6	'CBF7'	'CBF0'	'CBF1'	'CBF2'	'CBF3'	'CBF4'	'CBF5'	'CBF6'
7	'CBFF'	'CBF8'	'CBF9'	'CBFA'	'CBFB'	'CBFC'	'CBFD'	'CBFE'

Fig. 8A

BIT 5,E ; coloca o complemento do bit 5 de E no flag Z
JP Z endereço (CALL Z endereço)
(se a condição for que o bit seja 1, basta fazer JP NZ)

Se o bit 5 fosse 1, o complemento é Ø que é colocado no flag Z; desse modo, a condição JP Z testa se a flag Z está em 1 o que não ocorre e o salto não será efetuado. Em caso contrário (bit 5 em *zero*) o salto é efetuado.

Temos também instruções para setar o bit de CARRY e para complementá-lo, porém, note que não há uma instrução para resetar o CARRY (figura 11).

As instruções SET e RES não afetam nenhuma FLAG; as instruções BIT afetam todas as FLAGS menos o CARRY e as instruções SCF e CCF afetam apenas o CARRY.

Instruções de rotação: multiplicação e divisão por potências de dois

Iremos agora ver instruções que permitem fazer uma "rotação" dos registros ou memória indicada por (HL).

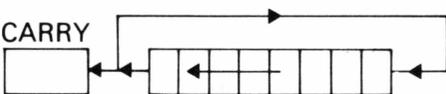
Algumas destas instruções podem ser utilizadas para multiplicar ou dividir *facilmente* algum número por um expoente de 2 (2, 4, 8, 16 etc.).

A primeira delas é a instrução RLC: ela desloca todos os bits para a *esquerda* num movimento de "rotação", ou seja, o bit 7 é colocado no bit Ø e os bits de Ø a 6 são deslocados de 1 posição para a esquerda; o bit 7 é copiado no CARRY; assim, por exemplo, se o registro D contiver.

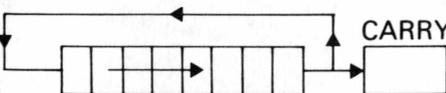
"1ØØ1ØØ1Ø e CARRY = Ø

ao efetuar RLC D teremos:

D = "ØØ1ØØ1Ø1" CARRY = 1

RLC

A instrução RRC é análoga, efetuando a rotação para a *direita*, copiando então o bit Ø no CARRY; assim se B = "11ØØØØ11" o CARRY = 1 ao efetuar RRC B teremos:

B = "111ØØØØ1" CARRY = 1

RRC

Temos também a instrução RL que executa uma rotação para a *esquerda* como se o registro (ou memória) e o CARRY formassem um único

RES

Bit \ Reg	A	B	C	D	E	H	L	(HL)
Ø	'CB87'	'CB8Ø'	'CB81'	'CB82'	'CB83'	'CB84'	'CB85'	'CB86'
1	'CB3F'	'CB88'	'CB89'	'CB8A'	'CB8B'	'CB8C'	'CB8D'	'CB8E'
2	'CB87'	'CB9Ø'	'CB91'	'CB92'	'CB93'	'CB94'	'CB95'	'CB96'
3	'CB9F'	'CB98'	'CB99'	'CB9A'	'CB9B'	'CB9C'	'CB9D'	'CB9E'
4	'CBA7'	'CBAØ'	'CBA1'	'CBA2'	'CBA3'	'CBA4'	'CBA5'	'CBA6'
5	'CBAF'	'CBA8'	'CBA9'	'CBAA'	'CBAB'	'CBAC'	'CBAD'	'CBAE'
6	'CBB7'	'CBBØ'	'CBB1'	'CBB2'	'CBB3'	'CBB4'	'CBB5'	'CBB6'
7	'CBBF'	'CBB8'	'CBB9'	'CBBA'	'CBBB'	'CBBC'	'CBBD'	'CBBE'

Fig. 8B

```
3ØØØØ   RES   6,B   'CBBØ'
3ØØØ2   RES   5,B   'CBA8'
3ØØØ4   RES   4,B   'CBAØ'
3ØØØ6   RES   2,B   'CB9Ø'
3ØØØ8   RES   Ø,B   'CB8Ø'
3ØØ1Ø   SET   7,C   'CBF9'
3ØØ12   SET   6,C   'CBF1'
3ØØ14   SET   3,C   'CBD9'
3ØØ16   SET   2,C   'CBD1'
3ØØ18   SET   1,C   'CBC9'
3ØØ2Ø   SET   Ø,C   'CBC1'
3ØØ22   RET         'C9'
```

Fig. 9

Bit \ Reg	A	B	C	D	E	H	L	(HL)
Ø	'CB47'	'CB4Ø'	'CB41'	'CB42'	'CB43'	'CB44'	'CB45'	'CB46'
1	'CB4F'	'CB48'	'CB49'	'CB4A'	'CB4B'	'CB4C'	'CB4D'	'CB4E'
2	'CB57'	'CB5Ø'	'CB51'	'CB51'	'CB52'	'CB53'	'CB54'	'CB55'
3	'CB5F'	'CB58'	'CB59'	'CB52'	'CB53'	'CB54'	'CB55'	'CB56'
4	'CB57'	'CB6Ø'	'CB61'	'CB62'	'CB63'	'CB64'	'CB65'	'CB66'
5	'CB6F'	'CB68'	'CB69'	'CB6A'	'CB6B'	'CB6C'	'CB6D'	'CB6E'
6	'CB77'	'CB7Ø'	'CB71'	'CB72'	'CB73'	'CB74'	'CB75'	'CB76'
7	'CB7F'	'CB78'	'CB79'	'CB7A'	'CB7B'	'CB7C'	'CB7D'	'CB7E'

Fig. 10

INSTRUÇÃO	CÓDIGO	
SCF	'37'	(SET CARRY flags)
CCF	'3F'	(complement carry flag)

Fig. 11

```
3ØØØØ   LD    B,Ø   'Ø6ØØ'   ; coloca Ø em B
3ØØØ2   SLA   C     '21'     ; multiplica C por dois
3ØØØ3   RET         'C9'
```

Fig. 12

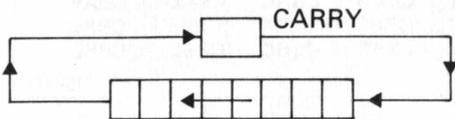
```
3ØØØØ   LD    B,Ø            ; coloca Ø em B
3ØØØ2   SRL   C     '39'     ; divide C por dois
3ØØØ3   RET
```

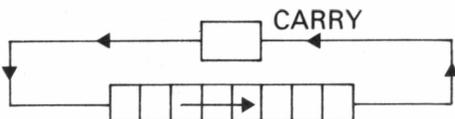
Fig. 13

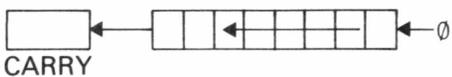
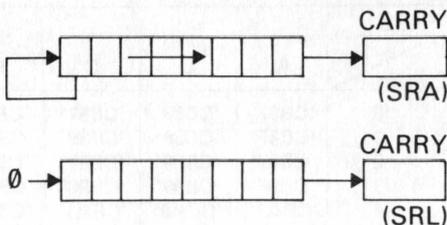
Inst. \ Reg	A	B	C	D	E	H	L	(HL)
RLC	'CBØ7'	'CBØØ'	'CBØ1'	'CBØ2'	'CBØ3'	'CBØ4'	'CBØ5'	'CBØ6'
RRC	'CBØF'	'CBØ8'	'CBØ9'	'CBØA'	'CBØB'	'CBØC'	'CBØD'	'CBØE'
RL	'CB17'	'CB1Ø'	'CB11'	'CB12'	'CB13'	'CB14'	'CB15'	'CB16'
RR	'CB1F'	'CB18'	'CB19'	'CB1A'	'CB1B'	'CB1C'	'CB1D'	'CB1E'
SLA	'CB27'	'CB2Ø'	'CB21'	'CB22'	'CB23'	'CB24'	'CB25'	'CB26'
SRA	'CB2F'	'CB28'	'CB29'	'CB2A'	'CB2B'	'CB2C'	'CB2D'	'CB2E'
SRL	'CB3F'	'CB38'	'CB39'	'CB3A'	'CB3B'	'CB3C'	'CB3D'	'CB3E'

Fig. 14

registro de 9 bits; ou seja, o CARRY é introduzido no BIT Ø, os BITS de Ø a 6 são deslocados de uma posição para a esquerda e o BIT 7 é colocado no CARRY; assim se C = "Ø1Ø1Ø1Ø1" e CARRY = 1, ao efetuar RL C teremos:

C = "1Ø1Ø1Ø11" e CARRY = Ø

A instrução RR é análoga, fazendo uma rotação para a *direita;* se A = "ØØØØ1111" e CARRY = Ø, ao efetuar RR A teremos:

A = "ØØØØØ111" e CARRY = 1

Vejamos agora a instrução SLA que permite efetuar a multiplicação por alguma potência de 2: ela desloca todos os bits para a esquerda e *reseta* o bit Ø; o bit 7 é copiado no CARRY. Analogamente à base 1Ø, quando multiplicamos um número por 1Ø, simplesmente deslocamos todos os algarismos para a esquerda e acrescentamos um zero no final! Assim ao executar SLA estaremos multiplicando o número por 2, ao executá-la duas vezes sucessivas por 4, 3 vezes por 8 e assim sucessivamente . . . Experimente a sub-rotina da figura 12 colocada a partir do endereço 3ØØØØ.

Ao fazer XF no HEXAMEM, o par BC é carregado com '753Ø'. Fazemos B = Ø e SLA C; ora C era: "ØØ11ØØØØ" (= 48) e se torna "Ø11ØØØØØ" (= 96 = 48 X 2). Assim obteremos 96 na tela: o fato do bit 7 ser colocado no CARRY nos permite detectar se houve um "estouro", ou seja, resultado maior do que 255.

Por analogia temos a instrução SRA que desloca os bits para a direita, SEM MODIFICAR o bit 7 e coloca o bit Ø no CARRY. Note entretanto que o fato do BIT 7 permanecer inalterado não permite "dividir" números positivos maiores do que 127 (="Ø1111111), mas possibilita também a divisão de números negativos (— 128); entretanto temos a instrução SRL que desloca os 8 bits para a direita, copia o bit Ø no CARRY e reseta o bit 7! com isto pode-se fazer a divisão de números positivos até 255 mas não é possível a divisão para números negativos (figura 13).

Ao executar este programa, devemos obter 24 (48 ÷ 2).

Na figura 14, vemos uma tabela com as instruções de rotação.

Instruções de rotação

Apenas a título de complementação, dê uma olhada na figura 15, onde vemos as algumas instruções.

INSTRUÇÃO	CÓDIGO
RLCA	'Ø7'
RRCA	'ØF'
RLA	'17'
RRA	'1F'

Fig. 15

Estas instruções produzem no acumulador o mesmo efeito que as instruções RLC A, RRC A, RL A e RRA, mas afetando *apenas* o flag de CARRY, sem afetar os demais FLAGS.

Exercícios

1. Usando as instruções vistas nesse capítulo, faça uma sub-rotina capaz de dividir o par de registros BC por 2; a seguir faça um programa que chame 2 vezes esta sub-rotina para dividir o conteúdo de BC por 4.

2. Faça um programa que multiplique o registro C por 7 (sem usar somas sucessivas) e detecte se houve um "estouro" (testando o CARRY).

3. Uma boa aproximação para o número π é 201/64. Faça um programa que, utilizando as sub-rotinas de multiplicação por somas sucessivas do capítulo anterior e a sub-rotina desenvolvida no exercício 1 multiplique o conteúdo do registro C por π, deixando o resultado em BC.

4. Faça um programa que forneça como saída o nº de bits em *um* de um dado registro; primeiramente, SEM utilizar instruções de rotação e, a seguir, outro programa que as utiliza.

5. Faça um programa que *tire a média* dos registros B e C (lembre-se que B + C pode ser maior que 255); a seguir, implemente o programa para que ele tire a média dos registros B, C, D e E.

Use as instruções de rotação para fazer as divisões. ○

O Merge para o TK explicado

Na edição número 10 de nossa revista publicamos um utilíssimo programa de MERGE para o TK 83/85 e seus compatíveis. Neste mês, recebemos uma colaboração que vem completar nosso artigo anterior. Esperemos que este artigo o auxilie na compreensão deste importante utilitário.

Daniel J. R. Nordemann

Introdução

O TK 83 e seus compatíveis oferecem a possibilidade de gravar um programa e dados numa fita cassete comum e de carregar a memória com um programa com dados já gravados. Mas, até agora, não existe a possibilidade de integrar da mesma maneira pacotes de dados já gravados a um programa presente na memória, ou de reunir dois programas diferentes em um só como faz a instrução MERGE existente em algumas máquinas. Não se trata da aquisição de dados, mas somente da sua manipulação por vários programas ou da integração de um programa a outro, usando apenas um gravador além do TK 83. O processo a ser descrito não precisa de modificação alguma nem de circuito complementar.

Manipulação dos Dados

A manipulação dos dados é fácil e não apresenta dificuldades como perda de memória ou perda de cursor. O que diremos a seguir aplica-se a qualquer dado numérico ou caractere, mas, para facilitar as explicações, trataremos dados sob a forma de bytes, sendo a generalização dos dados numéricos bastante evidente.

As mesmas posições disponíveis para programas escritos em hexadecimal podem ser usadas para os dados: dentro de um vetor de caracteres do tipo A$; num REM como 1 REM . . . cujo primeiro endereço útil é 16514; e acima de RAMTOP, após reservar um lugar conveniente abaixando RAMTOP através de instruções POKE, por exemplo, a seqüência de instruções POKE 16388, O (facultativa) e POKE 16389, 125 reserva 3 vezes 256 bytes (com extensão 16 k cujo RAMTOP "natural" é 32768 = 128*256). Chamar-se-ão 1, 2 e 3, respectivamente, os espaços na seqüência de caracteres, no primeiro REM e acima de RAMTOP. As transferências são operadas sem dificuldades, tal como exemplificado abaixo:

```
1—→2    FOR J = 1 TO 256
        LET V = 16513 + J
        POKE V, CODE A$ (J)
        NEXT J
```

Lembre-se que as variáveis estão localizadas entre VARS e E LINE depois do programa ter rodado:

TABELA I

CONTEÚDO DA MEMÓRIA			
Variável de uma letra A	byte 1	011. . . . letra A	64 + CODE "A"
Variável de algumas letras ABC	byte 1	101. . . . letra A	128 + CODE "A"
	byte 2	001. . . . letra B	CODE "B"
	byte 3	101. . . . letra C	128 + CODE "C"
Vetor numérico A ( )	byte 1	100. . . . letra A-20H	128 + CODE "A" − 32 = 96 + CODE "A"
Variável de controle de "loop" FOR-NEXT A	byte 1	111. . . . letra A	192 + CODE "A"
Seqüência de caracteres A$	byte 1	010. . . . letra A-20H	64 + CODE "A" − 32 = 32 + CODE "A"
Vetor de caracteres A$ ( )	byte 1	110. . . . letra A-20H	192 + CODE "A" − 32 = 160 + CODE "A"

Endereço de VARS = PEEK 16400 + 256*PEEK 16401
Endereço de E LINE = PEEK 16404 + 256*PEEK 16405

O manual de instrução mostra como são constituídas as diferentes variáveis (o que já foi interpretado e verificado experimentalmente a fim de apresentar a Tabela I, na qual as letras A, B, C são tomadas como exemplos).

Dispondo destas informações, pode-se escrever um pequeno programa de procura de variáveis pelo tipo e pelo nome, usando o valor de Z determinado graças à Tabela I (figura 1).

```
   5 LET A=1
   7 LET Z=102
  10 LET V=PEEK 16400+256*PEEK 1
6401
  20 FOR J=0 TO 1E3
  30 LET W=V+J
  40 IF PEEK W=2 THEN GOTO 80
  50 NEXT J
  60 PRINT "NAO ACHEI"
  70 STOP
  80 FOR J=0 TO 20
  90 LET Y=W+J
 100 PRINT Y;" ";PEEK Y
 110 NEXT J
 190 STOP
```

Fig. 1

A linha 5 seria uma das escolhidas de acordo com a tabela II.

TABELA II

Variável	Linhas:	Z =
Variável de uma letra	5 LET A = 1	102
Variável de algumas letras	5 LET ABC = 1	166
Tabela numérica	5 DIM A(10)	134
Variável de controle de "loop"	5 FOR A = 1 TO 9	230
Seqüência de caracteres	5 LET A$ = "STAR"	70
Tabela de caracteres	5 DIM A$(5, 10)	198

Até agora não foi esclarecido como um dado numérico preenche as cinco casas de memória, ou seja, os cinco bytes a ele atribuídos. No caso dos caracteres, é bem mais fácil, pois cada byte representa o código CODE do caractere correspondente. O programa da figura 1 explica, melhor do que qualquer fórmula, a correspondência entre os valores numéricos e os cinco bytes que eles ocupam na memória.

```
   5 PRINT "A ?"
   6 INPUT A
   8 FAST
  10 LET V=PEEK 16400+256*PEEK 1
6401
  20 FOR J=0 TO 1E3
  30 LET W=V+J
  40 IF PEEK W=102 THEN GOTO 55
  50 NEXT J
  55 CLS
  60 SLOW
  70 PRINT "A= ";A
  75 PRINT
  80 FOR J=0 TO 5
  90 LET Y=W+J
 100 PRINT ;" ";PEEK Y
 110 NEXT J
 130 LET P=0
 140 IF A THEN LET P=INT (LN ABS
 A/LN 2)
 150 LET N=128*(ABS A/2**P-1)
 160 IF A THEN LET P=P+129
 170 LET Q=128*(A<=0)+INT N
 180 LET N=N-INT N
 190 LET R=INT (256*N)
 200 LET N=256*N-R
 210 LET S=INT (256*N)
 220 LET T=INT (256*N-S)
 230 PRINT "P= ";P
 240 PRINT "Q= ";Q
 250 PRINT "R= ";R
 260 PRINT "S= ";S
 270 PRINT "T= ";T
 290 STOP
```

Fig. 2

Os valores Z (na linha 7) foram determinados de acordo com a Tabela 1, levando em conta o valor de CODE"A = 38. Lembre-se que CODE "B" = 39, CODE "C" = 40 etc. No caso das variáveis definidas por mais de um caractere, o primeiro tem de ser uma letra. Se o programa apresenta algumas variáveis cujo nome começa pela mesma letra, a identificação tem de usar mais letras (ou todas, se quiser), modificando a linha 40 assim:

```
  40 IF PEEK W=Z1 AND PEEK (W+1)
=Z2 AND PEEK (W+2)=Z3 THEN GOTO
80
```

Fig. 3

com

Z1 = 128 + CODE "A" (ou primeira letra)
Z2 = CODE "B" . . . (ou segunda letra)
Z3 = 128 + CODE "C" (ou última letra)

e somente os códigos da primeira e da última letra são acrescentados de 128.

O programa prevê a possibilidade de não achar o endereço do nome da variável, o que pode acontecer caso haja discordância entre nome, tipo e valor de Z. Se o endereço for achado, a seqüência seguinte do programa (linha 80 e seguintes) visualiza a estrutura da variável e permite conferir com as informações (até agora pouco usadas) que são fornecidas pelo manual de instrução. Esta estrutura foi lembrada em termos resumidos. O programa dado acima permite verificar os elementos relativos ao nome da variável, ao valor de Z e ao dimensionamento.

Em função do tipo de variável, as seqüências dos bytes são as da Tabela III.

TABELA III

Variável de uma letra	Z/valor 5 bytes/
Variável de algumas letras	Z1/Z2/Z3/valor de 5 bytes
Vetor numérico	Z/comprimento em 2 bytes: elementos + 2 por dimensão + 1 para número de dimensões/primeira dimensão 2 bytes/ . . ./ . . ./última dimensão 2 bytes/valores 5 bytes cada/
Variável de controle de "loop"	Z/valor 5 bytes/limite 5 bytes/step 5 bytes/linha FOR-NEXT + 1: 2 bytes/
Seqüência de caracteres	Z/número de car. 2 bytes/conteúdo 1 byte por car./
Vetor de caracteres	Z/comprimento em 2 bytes: elementos + 2 por dimensão + 1 para número de dimensões/primeira dimensão 2 bytes/ . . ./última dimensão 2 bytes/conteúdo 1 byte por car./

Quando o comprimento, ou uma dimensão, é colocado em dois bytes, o primeiro é o menos significativo e o segundo o mais significativo.

O cálculo dos bytes P, Q, R, S e T esclarece como é armazenando um dado numérico em cinco bytes de memória. P representa o número de potências de 2, ou seja, a característica do número expresso nos logaritmos à base 2. A parte fracionária constitui a mantissa e é distribuída nos quatro bytes seguintes, usando os sete bits menos significativos do primeiro byte e nos três bytes restantes R, S e T. O bit mais significativo de Q é atribuído ao sinal de A; com 1 para $A < 0$ e 0 para $A \geqslant 0$. A fórmula da linha 150 entende-se mais facilmente quando é escrita da seguinte maneira:

N = 128* (ABS A — 2**P) / (2**(P+1) — 2**P)

que se simplifica na fórmula da linha 150 após dividir numerador e denominador por 2**P. As condições sobre A<>0 nas linhas 140 e 160 justificam-se para não haver problema com o logaritmo e para obter 0 para P, Q, R, S e T quando A = 0.

Estas explicações podem não parecer bastante claras, mas as fórmulas que dão P, Q, R, S e T contêm as informações necessárias ao entendimento da estrutura dos dados numéricos na memória, sob forma de cinco bytes. No outro sentido, a fórmula seguinte, escrita também em termos de BASIC, mostra que se pode passar dos cinco bytes ao dado numérico no sistema decimal:

```
 275 LET B=(1-2*(Q>128))*(2**(P-
129))*(1+(Q-128*(Q>128))/128+R/3
2768+S/8388608+T/8388608/256)
 280 PRINT
 285 PRINT "B= ";B
```

Fig. 4

O termo 1 — 2*(Q > 128), que fornece o sinal de B, foi posto no lugar de (— 1)**(Q > 128), uma vez que o TK83/85 se recusa a dar as potências de números negativos.

Agora, com o conhecimento dos endereços e da estrutura das variáveis numéricas e das seqüências de caracteres, existe a possibilidade de "POKEar" qualquer coisa nelas e transferir dados pelas instruções LD em hexadecimal. Uma recomendação importante ainda: a cada modificação de programa esses endereços vão sendo alterados; os dados não são perdidos, porém há que procurá-los no lugar certo, usando o programa que foi dado acima.

O próximo artigo mostrará como integrar um programa (gravado) a outro, já na memória da máquina, e como tratar dados quaisquer por um programa ao qual eles não estavam associados.

Referências Bibliográficas

MOREIRA, C.H. Empacotamento de string. *Nova eletrônica*, 6 (74): 77-80, abr. 1983.

PROGRAMA-PADRE

ou "Casando impedâncias por computador"

Tanios Hamzo

No último número de MICRO-HOBBY, apresentamos um programa capaz de projetar amplificadores. Esperando não estar insistindo na mesma tecla, apresentamos este mês um outro programinha simples e muito útil: O PROGRAMA-PADRE, destinado a calcular rápida e precisamente uma rede resistiva do tipo "L", fornecendo o valor dos resistores, a perda de tensão, em deciBéis e fornecendo o esquema elétrico, que é desenhado na tela e pode ser reproduzido em papel via impressora.

Ainda dentro da filosofia T.A.C. (Trabalho Auxiliado por Computador) — ou, se preferir: C.A.D. (Computer-Aided Designs), este programa é uma verdadeira M.N.R. (Mão-Na-Roda) para estudantes, engenheiros, técnicos e eletrocuriosos em geral.

Embora ciente de ser esta uma revista dedicada a computadores do tipo do TK, optei por desenvolver programas aplicativos e didáticos para computadores da família T.R.S. (Tandy-Radio-Shack), dada sua popularização. Estão aí os TRS-80, os CP-500, os CP-300, os Dismac série D-8000, etc. sendo usados em lares, escolas, faculdades e empresas e acessíveis a uma grande parte de usuários do TK. Para aqueles que ainda não tiveram contato com um "oitenta" e acham que estão sendo prejudicados por esta seção, um lembrete: Não se iluda — mais cedo ou mais tarde, um "oitenta" ou um pomo da macieira Apple irá cruzar o seu caminho e então o que "Tkaiano" iria fazer? Para verdadeiros programadores, o que importa é o programa, e não meras sintaxes de diferentes fabricantes.

Para ajudar aqueles que não desejam bitolar seu raciocínio sintático usando exclusivamente um mesmo tipo de computador, "traduzi" o programa-padre, desenvolvido inicialmente para um TRS-80 model IV para a sintaxe do TK. Só espero não estar, desta forma, prejudicando agora aqueles que têm acesso a um "oitenta' . . .

A listagem 1, mostra o programa original e a figura 1, o esquema do circuito casador resistivo de impedâncias do tipo "L".

Observe a simplicidade do circuito; dois resistores ligados em rede "L" entre as linhas de diferentes impedâncias, "casando" a maior impedância presente na entrada com a menor impedância presente na saída até que a morte (ou uma necessidade de projeto) os separe . . .

Ao desenvolver o programa, minha principal preocupação foi a de complicar o mínimo possível. A escolha de um casador de impedâncias usando exclusivamente resistores, implicou em:

- Cálculo simples, sem se recorrer a fórmulas com números imaginários (os números complexos) nem levar em conta a freqüência.
- Didático; fácil de se entender.
- Montagem rápida e simples do circuito.
- Custo reduzido dos componentes.

Um pouco de teoria

Impedância pode ser entendida como sendo uma medida de resistência para corrente alternada. Dependendo da freqüência, capacitores e indutores apresentam diferentes impedâncias. A impedância pode ser expressa como sendo:

$$\text{IMPEDÂNCIA} = \frac{\text{Tensão em corrente alternada no circuito}}{\text{Corrente alternada do circuito}}$$

Em outras palavras (ou fórmulas):

$$\text{RESISTÊNCIA} = \frac{\text{Tensão}}{\text{Corrente}}$$

no caso de se tratar de corrente contínua.

Um resistor ôhmico puro tem a mesma resistência em corrente contínua e em corrente alternada, independentemente da freqüência.

Já um capacitor, se comporta, teoricamente como um interruptor desligado em corrente contínua; tem sua impedância inversamente proporcional à freqüência: ou seja, quanto maior a freqüência, menor a impedância, conforme a fórmula:

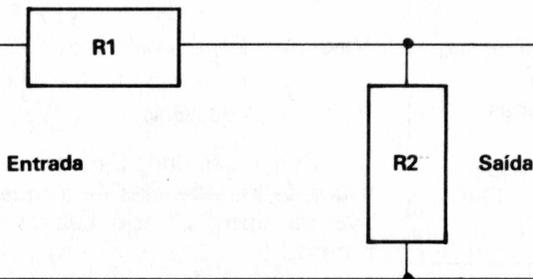
$$\text{IMPEDÂNCIA DO CAPACITOR} = \frac{1}{\text{FREQÜÊNCIA} \times \text{CAPACITÂNCIA} \times \sqrt{-1}}$$

Sendo $\sqrt{-1}$ = i, número imaginário.*

Um indutor, por sua vez, comporta-se teoricamente como um interruptor ligado quando submetido a uma corrente contínua. Em corrente alternada, a impedância de um indutor é diretamente proporcional à freqüência; isto é: quanto maior a freqüência, maior a impedância apresentada pelo indutor, conforme a fórmula:

$$\text{IMPEDÂNCIA DO INDUTOR} = \text{FREQÜÊNCIA} \times \text{INDUTÂNCIA} \times \sqrt{-1}$$

*** Nota do Editor: Se você desconhecer os números complexos, consulte um livro de matemática de curso colegial (2º grau).**

PROGRAMA ORIGINAL

```
10 CLS
20 PRINT"CASADOR DE IMPEDANCIAS - Red
e tipo 'L'"
30 INPUT"Qual a impedancia de entrada
";E
40 INPUT"Qual a impedancia de  saida
";S
50 R1=E*SQR(1-(S/E))
60 R2=S*SQR(1-(S/E))
70 M=SQR(E/S)+SQR((E/S)-1)
80 P=20*(LOG(M)/LOG(10))
90 CLS
100 PRINT@514,"ENTRADA"
110 PRINT@143,"R1 = ";R1;" ohms"
120 PRINT@560,"SAIDA"
130 PRINT@920,"Perda : ";P;"  dB"
140 PRINT@591,"R2 = ";R2;" ohms"
150 Y=15:C=10:F=30:GOSUB190:C=58:F=10
0:GOSUB190:Y=12:C=30:F=58:GOSUB190:Y=
18:GOSUB190:Y=21:C=74:F=87:GOSUB190:Y
=32:GOSUB190:Y=36:C=10:F=100:GOSUB190
160 X=30:C=12:F=18:GOSUB200:X=58:GOSU
B200:X=80:C=15:F=21:GOSUB200:C=32:F=3
6:GOSUB200:X=74:C=21:F=32:GOSUB200:X=
87:GOSUB200
170 PRINT@0,"";:INPUT"Deseja recomeca
r o programa (S/N) ";R$
180 IF R$="S" THEN GOTO 10 ELSE THEN
GOTO 180
190 FOR X=CTOF:SET(X,Y):NEXTX:RETURN
200 FOR Y=CTOF:SET(X,Y):NEXTY:RETURN
```

PROGRAMA ADAPTADO PARA O TK

```
   1 REM TANIOS HAMZO
   5 POKE 16510,0
  10 PRINT ,,"CASADOR DE IMPEDAN
CIAS/ REDE ""L"""
  20 PRINT ,,,,"QUAL A IMPEDANCI
A DE ENTRADA?"
  30 INPUT E
  40 PRINT ,,"QUAL A IMPEDANCIA
DE SAIDA?"
  50 INPUT S
  60 LET R1=E*SQR (1-(S/E))
  70 LET R2=S*SQR (1-(S/E))
  80 LET M=SQR (E/S)+SQR ((E/S)-
1)
  90 LET P=20*(LN (M)/LN (10))
 100 CLS
 110 PRINT AT 6,6;"▪▪▪ R1=";INT
(R1);" OHMS";AT 7,6;"▪▪▪";AT 11,
0;"ENTRADA:  R2=";INT (R2);" OHM
S";TAB 22;"▪▪  SAIDA:";E;" OHMS"
;TAB 22;"▪▪";S;" OHMS";TAB 22;"▪
▪       PERDA:";INT (P);" DBS"
 120 FOR Y=9 TO 29 STEP 20
 130 FOR X=0 TO 63
 140 PLOT X,Y
 150 NEXT X
 160 NEXT Y
 170 LET X=45
 180 FOR Y=10 TO 28
 190 PLOT X,Y
 200 NEXT Y
```

COMENTÁRIOS SOBRE O PROGRAMA (listagem 1):

LINHA:	***DESCRIÇÃO:***
30 e 40	Entrada de dados.
50	Cálculo do resistor R 1 , em ohms.
60	Cálculo do resistor R 2 , em ohms.
70 e 80	Cálculo de perda de tensão, em deciBéis.
100 a 140	Apresentação dos resultados e denominação do esquema.
170	Pergunta pelo recomeço.
180	Decisão entre o recomeço ou trava do programa.

DESENHO DO ESQUEMA:

150	Coordenadas das linhas horizontais.
160	Coordenadas das linhas verticais.
190	Sub-rotina para desenhar as linhas horizontais.
200	Sub-rotina para desenhar as linhas verticais.

A listagem 2 mostra o programa-padre adaptado para o TK. Note que não há diferenças radicais se comparado à listagem 1. Há apenas três diferenças quanto à sintaxe: o @ (AT, no TK), o SET (PLOT, no TK) e o ELSE, que não tem tradução simples para o TK, mas que pode facilmente ser substituído por outro procedimento na programação.

Exemplo de Aplicação:

Deseja-se casar a impedância de uma linha com 300 ohms com 75 ohms de outra linha. Roda-se então o programa-padre para unir os laços (de corrente, não matrimoniais) de linhas com impedâncias diferentes e obtém-se os seguintes resultados:

Valor de R 1 : calculado: 260 ohms
prático: 270 ohms

Valor de R 2 : calculado: 65 ohms
prático: 68 ohms

Perda: 11,4 deciBéis

Com estes dados, monta-se o casador de impedâncias — a rede resistiva do tipo "L" (de Lua-de-mel) e pronto. O

VIDEOCOMPO
National
Scotch
C-20
Computer Cassette
3M
3,81 mm x 30,5 m
Recomendado para uso até 2000 BPS.
Programações mais seguras.
• A **única** fita com o comprimento especialmente dimensionado para 20 minutos.
• Evita a regravação de programas.
• Elimina a ocorrência de **dropouts.**
• Requer um torque mais baixo do gravador.
• Proporciona maiores ganhos nas frequências médias e altas.
NOVA
Seus programas merecem o Computer Cassette C-20 Scotch!
A fita da mais alta segurança, com a qualidade Scotch.
Produtos Magnéticos CPD/3M
Caixa Postal 123 - Campinas - SP
3M

Aqui você tem a melhor iniciação em microcomputação que existe.

O TK 83 já ensinou mais de 2 milhões de pessoas. Ele é muito fácil de operar. Usa o Basic, e a memória chega até 64 K bytes, e aceita monitor, impressora e joystick.

Num instante você vai estar resolvendo problemas programando, ou vencendo os muitos jogos disponíveis. O TK 83 não é só a melhor iniciação. Também é a mais divertida.

Aqui você já aplica os seus conhecimentos

Com o TK 85 você também pode se divertir muito: ele tem dezenas de jogos disponíveis.

Mas ele já é mais sofisticado. Tem software já pronto. Linguagens Basic e Assembler. Teclado tipo máquina de escrever, com 40 teclas e 160 funções. 16 ou 48 K de memória RAM, e 10 de ROM. Gravação em high-speed, e função Verify, para maior segurança.

Quando você já estiver apaixonado por microcomputação, ele vai corresponder totalmente.

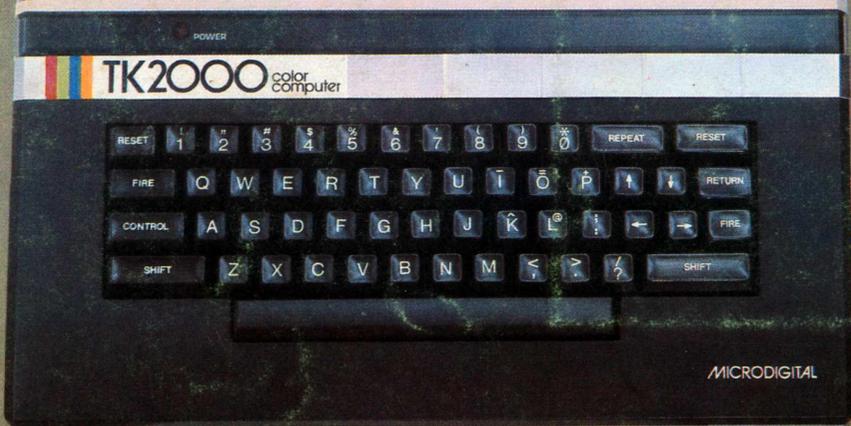
Aqui você mostra tudo o que sabe.

O TK 2000 Color tem tudo que os melhores micros têm. Menos o preço. Aceita diskette, impressora (já vem com interface), alta resolução gráfica à cores podendo ser ligado ao seu TV colorido ou P&B. Tem 64 k de memória RAM e 16 k de memória ROM. Com excelente software disponível.

Você pode mostrar tudo o que sabe. Sem precisar mostrar muito dinheiro.

MICRODIGITAL

Microdigital Eletrônica Ltda
Caixa Postal 54121 - CEP 01000 -
São Paulo - SP Telex nº (011) 37008 MIDE BR

* Preço sujeito a alteração

À venda nas boas casas do ramo, lojas especializadas de fotovideo-som e grandes magazines em: **ALAGOAS** - Maceió, Palmeira dos Indios, **AMAZONAS** - Manaus, **BAHIA** - Salvador, **CEARÁ** - Fortaleza, **DISTRITO FEDERAL** - Brasília, **ESPÍRITO SANTO** - Vitória, **GOIÁS** - Goiânia, **MATO GROSSO** - Cuiabá, **MINAS GERAIS** - Belo Horizonte, Divinópolis, Itajuba, Juiz de Fora, Poços de Caldas, São João Del Rei, Teófilo Otoni, Uberlândia, Uberaba, Viçosa, **PARAÍBA** - Campina Grande, **PARÁ** - Belém, **PARANÁ** - Curitiba, Londrina, Maringá, **PERNAMBUCO** - Recife, **RIO DE JANEIRO** - Campos, Niterói, Nova Friburgo, Petrópolis, Rezende, Rio de Janeiro, Volta Redonda, **RIO GRANDE DO SUL** - Bagé, Canoas, Caxias do Sul, Ijuí, Novo Hamburgo, Pelotas, Porto Alegre, Sant'Anna do Livramento, Santiago, Santa Rosa, São Leopoldo, **RIO GRANDE DO NORTE** - Natal, **RONDÔNIA** - Porto Velho, **SÃO PAULO** - Araraquara, Assis, Avaré, Bauru, Birigui, Botucatu, Campinas, Catanduva, Franca, Guarulhos, Itu, Jacarei, Jaú, Limeira, Lins, Marilia, Mogi Guaçu, Mogi das Cruzes, Ourinhos, Piracicaba, Pirassununga, Promissão, Rio Claro, Ribeirão Preto, Santos, Santa Barb. d'Oeste, São Bernardo do Campo, São João da Boa Vista, São Sebs. da Grama, São Carlos, São José do Rio Preto, São José dos Campos, Stº André, São Paulo, Sorocaba, Suzano, Taubaté, **SANTA CATARINA** - Blumenau, Brusque, Florinópolis, Itajaí, Joinville, **SERGIPE** - Aracajú, Se você não encontrar este equipamento na sua cidade ligue para (011) 800 - 255.8583.